Títol: **Plataforma d'autenticació modular per a Microsoft Windows**

Volum: **1/1**

Alumne: **Sergi Morales Surribas**

45494083 – Z

Director/Ponent: **Juan Carlos Cruellas Ibarz**

Departament: **Arquitectura de computadors**

Data:

DADES DEL PROJECTE

Títol del Projecte: Plataforma d'autenticació modular per a Microsoft Windows

Nom de l'estudiant: Sergi Morales Surribas, 45494083-Z

Titulació: Enginyeria Informàtica

Crèdits: 30

Director/Ponent: Juan Carlos Cruellas Ibarz

Departament: Arquitectura de Computadors

MEMBRES DEL TRIBUNAL *(nom i signatura)*

President:

Vocal:

Secretari:

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

Índex

RESUM EXECUTIU

Aquest capítol pretén donar una visió ràpida de perquè s'ha realitzat el projecte i quin n'ha estat el resultat.

Resum

La idea de realitzar el projecte comença al adonar-nos que els entorns informàtics homogenis tenen poc temps de vida i que cada vegada les organitzacions requereixen de diverses solucions on la òptima en la relació cost/funcionalitat no necessariament està disponible en el mateix entorn que tenen implantat.

Una de les reticències més grans que citen els departaments de Tecnologies de la Informació per implantar nous sistemes, és com integrar els diversos sistemes d'autenticació per tal poder continuar exigint la mateixa política de contrasenyes coherent amb les línies marcades per direcció, i que tota persona que ho requereixi disposi del mateix nom d'usuari i a ser possible la mateixa paraula de pas per tal de facilitar l'ús de les eines als usuaris. En diversos estudis es demostra que passar d'un entorn amb diverses credencials per persona cap a un entorn amb només una credencial, aporta uns estalvis de 132€/usuari/any en suport i pèrdues de productivitat, un import gens menyspreable en organitzacions de mida considerable.

Per tant, el que s'ha realitzat ha estat un estudi de quines opcions disposàvem al mercat, quines estratègies d'implementació teníem, d'acord amb el que recomana Microsoft o hem vist que realitzaven les solucions al mercat, i prendre una decisió sobre quina podia ser la millor solució.

La decisió presa, portar els *Pluggable Autentication Modules* (PAM) cap a l'entorn Windows, té força sentit si tenim en compte que en el món dels sistemes oberts (Unix) i a causa de la seva forma de treballar basada en diverses petites aplicacions sense molta relació entre elles i per tant independents a nivell de codi, es van adonar que continuar mantenint el codi dels diversos proveïdors d'autenticació a cada aplicació i els problemes d'introduir un nou sistema d'autenticació eren massa importants i acabarien penalitzant l'estabilitat, la capacitat d'evolució i la facilitat de gestió de l'entorn. Per tant, van decidir treure l'autenticació de les aplicacions i passar-la a gestionar de forma centralitzada amb el que van anomenar PAM, així el servidor de Telnet, FTP i SSH, podien compartir el proveïdor d'autenticació encara que els desenvolupadors dels servidors no sabessin que existia, donant a l'administrador del sistema el màxim de llibertat al gestionar els seus usuaris.

Un cop sabíem com implementar-ho, tocava entendre més a fons les necessitats dels administradors, desenvolupadors d'aplicacions i desenvolupadors de mòduls. Al tenir cada un les seves pròpies necessitats i entendre que si algú volia explotar comercialment el resultat del PFC, també havíem de tenir en compte les seves necessitats, les vam incloure i el resultat ha estat la necessitat d'un equilibri entre tots, per tal d'aconseguir una massa crítica suficient que n'expandeixi l'adopció.

Definits els requeriments, s'ha especificat, dissenyat i implementat la solució, desenvolupat aplicacions i mòduls d'exemple i comprovació i escrit un manual per administradors així com per a desenvolupadors.

INTRODUCCIÓ

Definició del projecte i organització de la memòria.

Definició del Projecte

Anàlisi d'alternatives, especificació, disseny i desenvolupament d'una Plataforma d'Autenticació Modular per a Microsoft Windows que permeti que el procés d'autenticació de l'usuari es pugui dur a terme contra servidors no Windows. Desenvolupament d'eines i exemples per poder desenvolupar noves aplicacions i mòduls que aprofitin la Plataforma.

Descripció del Projecte

Els sistemes operatius de Microsoft basats en la tecnologia NT resolen l'autenticació d'usuaris a través dels següents paquets de seguretat:

- dominis NTLM en el cas de Windows NT 3.x – 4.0 [MS10]

- *Active Directory* (Kerberos [MIT01] amb Ldap [RFC4510]) en Windows 2000 i Windows 2003.

Tot i que en els dominis amb *Active Directory* [MS07] tenim l'opció de configurar relacions de confiança amb reialmes Kerberos V no-Windows, perdem molts dels avantatges del l'*Active Directory* i per tant només s'hauria d'utilitzar en casos extrems: com per exemple autenticació en una Intranet (recurs usat) que corri sobre Windows 2000, però que els usuaris formin part d'un reialme Kerberos no-Windows.

A més, per tal de poder interoperar amb altres reialmes Kerberos s'ha de tenir un *Active Directory*, ja que les confiances s'estableixen a nivell de reialmes i no de client/reialme. És per això que per autenticar contra Kerberos es necessita una amplia infraestructura (controlador de domini, DNS propi etc.) si volem utilitzar Windows en un entorn Kerberos no basat en Windows.

Què és Kerberos?:

Kerberos és un mètode d'autenticació de requestes a serveis. El protocol s'ha desenvolupat al MIT (Massachussets Institute of Technology) i actualment estem a la versió 5. El nom prové del gos amb tres caps que protegia les Portes d'Hades de la mitologia grega, ja que Kerberos funciona mitjançant tres actors. El que fa la petició, el que ofereix el servei i el que autoritza al peticionari davant del que ofereix el servei. [MIT01]

Així, el que es necessita és algun *software* que permeti autenticar contra diversos proveïdors d'autenticació (no només NTLM i Kerberos) i que a més a més no sigui gaire exigent en quant a requeriments de màquina, infraestructura i administració.

Per tant, la solució passa per un programari que s'executi molt a prop de l'aplicació que necessita autenticar a l'usuari. Ha de ser capaç de tenir diverses configuracions per diversos aplicatius i que a més sigui possible afegir nous proveïdors d'autenticació sense gaire esforç.

Organització de la Memòria

En aquesta secció explicarem que inclou cada capítol per tal d'orientar ràpidament al lector.

Informe de definició

Aquest capítol s'explica el perquè i el quan del projecte.

- Antecedents: Explica com hem arribat a proposar aquest projecte.
- Motivació: Perquè creiem que té sentit realitzar un projecte com aquest.
- Com autentica Windows: Descripció dels mètodes d'autenticació disponibles per a Windows NT/2000/XP/2003.
- Com s'acostumen a gestionar els usuaris en Windows: Descripció dels mètodes més comuns d'implementació de la gestió d'usuaris en l'entorn Windows.
- Planificació del Projecte: Planificació en temps i recursos per tal de dur a terme el projecte.

Alternatives

En aquest capítol s'estudien diverses alternatives disponibles, tant per a Windows com per a d'altres plataformes.

- NtLdap: Treball dirigit, anterior a aquest projecte.
- pGina: Projecte de la Pacific Lutheran University basat en una *GInA* alternativa.
- Novell SecureLogin: Plataforma de Novell de Single-Sign-On.
- PAM: *Pluggable Authentication Modules*, permeten definir mòduls pels diversos proveïdors d'autenticació. Només estan disponibles en Unix/Linux.

Opcions

Aquí s'expliquen les diverses opcions que tenim per proveir d'un entorn flexible tant a desenvolupadors com a administradors de sistemes.

- Security Packages: API de Microsoft per implementar proveïdors de seguretat.
- Mòduls monolítics: Cada mòdul implementa tota la gestió i comunicació amb els diversos proveïdors d'autenticació.
- *GInA* amb Security Packages: Recomanació de Microsoft i la que segueixen la majoria d'implementacions comercials.
- PAM: Portar els PAM a l'entorn Windows adaptant-los a les particularitats d'aquest.
- Elecció: Quina opció triem.

Anàlisi de Requeriments

En aquest capítol estudiem quins requeriments tenim tant per altres desenvolupadors de mòduls i d'aplicacions, com per administradors de sistemes. Ja que són ells els qui hauran d'implantar les solucions desenvolupades amb la PAM^{2}w.

Especificació

Aquí especifiquem com interactuaran les diverses parts del sistema, quins drets i deures tenen cadascuna.

Disseny

Com serà l'arquitectura interna de cada una de les parts per tal d'acomplir els requisits del capítol anterior.

Eines i Metodologia d'Implementació

En aquest capítol discutirem de quines eines, metodologies i tecnologies disposem per implementar la solució a la que hem arribat en el disseny.

Conclusió

Aquí realitzarem un balanç del Projecte mitjançant quatre seccions:

- Objectius Assolits: Relació dels objectius assolits i quins, per poc prioritaris o per excessivament costosos, s'han deixat com a línia oberta.
- Recursos usats: Divergències entre la planificació inicial i la definitiva.
- Línies de treball obertes: Possibles millores o avanços aprofitant el desenvolupat en aquest projecte.
- Valoració personal: Quina finalitat ha tingut el Projecte per mi a nivell personal i tècnic.

Apèndix A

En aquest apèndix annexem el contingut de l'ajuda de l'equip de desenvolupament de *software* (SDK) de la PAM^{2}w.

Apèndix B

S'hi inclou el manual per a l'administrador de sistemes a fi de poder instal·lar, configurar i resoldre problemes amb la PAM^{2}w.

Motivació Personal

Al acabar el treball dirigit (*v. Antecedents*) vaig tenir el desig d'arribar més lluny del que havíem assolit. Permetre que una màquina Windows es comportés correctament en un context heterogeni, on poder treballar sense que tot l'entorn s'hagués d'emmotllar als requeriments de Microsoft. Així una organització que treballi en àmbit no-Windows pot aprofitar tot el software i eines disponibles en aquest sistema operatiu sense necessitat d'establir-hi una illa Microsoft.

L'objectiu del treball dirigit era permetre que els usuaris gestionats amb proveïdors d'autenticació Ldap no-Windows, poguessin entrar localment a la màquina Windows per mitjà de la sincronització de comptes d'usuari. El projecte que ens ocupa, no només permet el que hem esmentat del treball dirigit, si no que fa possible utilitzar diferents proveïdors d'autenticació i aporta moltes més funcionalitats que les d'accés interactiu a la màquina. Com ara:

- Servidors de BBDD

- Servidors FTP

- Web, etc.

Durant la meva estada al Laboratori de Càlcul de la FIB (LCFIB), m'encarregava de l'entorn Windows. Cada vegada hi havia més necessitat de proveir serveis basats en aquest entorn. Per exemple a l'assignatura de Disseny de Base de Dades, es donava docència sobre Microsoft SQLServer [MS06], i a l'entorn de la FIB, els proveïdors d'autenticació eren servidors Ldap [RFC4510] corrent sobre Solaris [SM01]. És per això que vaig decidir portar a terme aquest projecte a fi de facilitar la feina dels administrador de sistemes que es trobin en casos semblants.

A part, endinsar-se a les entranyes de Windows és per a mi un repte, per, la falta de formació a la carrera en tecnologies Microsoft, i la dificultat que té trobar informació editada per terceres persones sobre els aspectes interns dels productes d'aquesta multinacional. No se solen trobar articles que facin referència a aquesta empresa, o que expliquin com modificar el comportament de Windows. Malgrat aquest handicap, Microsoft disposa d'un una documentació que qualificaria de correcte. De fet, supera en documentació pròpia a molts d'altres fabricants i projectes, tot i això provoca que a vegades s'hagi d'investigar amb el *debugger* o mitjançant modificacions en el codi del programa, per tal de veure quin és el seu comportament en execució (*xivatos*) en cas de no trobar la informació desitjada en la documentació.

Per últim, crec que aquest projecte pot tenir una esperançadora continuïtat. Per desgracia no per part meva, ja que la feina i càrrec que m'ocupen no m'ho permeten. És per això mateix, que tinc la intenció de publicar-ho en breu dins de la comunitat de *Opensource* de manera a que ells el puguin llençar al mercat. Per tant, el llicenciarem segons la *Open Source Initiative* [OSI01]. D'altre banda, tinc la creença, i esperança a l'hora, de que aquest producte pugui ser en un futur no gaire llunyà un element necessari pel desenvolupament en el món Microsoft. De fet com es veurà en la motivació del projecte el cost de la implantació d'una solució basada en PAM^{2}w pot ser força elevat sense allargar gaire el retorn de l' inversió.

INFORME DE DEFINICIÓ

D'on ve el projecte, perquè i com el farem.

Antecedents

El projecte sorgí després de la realització, per part meva, d'un treball dirigit per Leandro Navarro (leandro *at* ac.upc.es). En ell es desenvolupà un substitut per la Windows *Graphical Identification and Authentication (GInA)* [MS11]. Aquest treball dirigit fou la resposta a una necessitat del Laboratori de Càlcul de la FIB [LCF01]. S'havia d'integrar les màquines Windows (estacions de treball i servidors) en un entorn majoritàriament Unix, on la gestió de comptes d'usuaris es realitzava en servidors Ldap (*iPlanet Directory Server*).

El treball, consistí en l'estudi de les diverses opcions que Microsoft proporcionava per tal de proveir un sistema d'autenticació alternatiu a l'accés interactiu a màquines Windows NT/2000/XP. Després d'estudiar-ho, es veié que totes les opcions passaven pel desenvolupament d'un substitut de la *GInA.* Per qüestió de temps, optàrem per limitar-nos a desenvolupar una *GInA* que autentiqués primer els usuaris amb el servidor Ldap. En cas d'èxit, la nostra *GInA* intentarà autenticar-los en la màquina local, creant, si no hi era ja, un compte local per a l'usuari. Aleshores es passarà a l'entorn de l'usuari i es carregarà l'escriptori (*shell*).

Un cop resolt el problema de l'accés interactiu a la màquina, s'observà que també fóra útil proporcionar un sistema flexible d'autenticació, no només a l'accés interactiu, sinó també a diverses aplicacions amb funció de servidor, que s'executen sobre la plataforma Windows.

Motivació

Aquest projecte es planteja en el context d'una organització que no ha estandarditzat en un determinat entorn de sistemes de la informació. Per tant, el que tenim són diverses illes on les aplicacions que treballen en el mateix entorn (Microsoft Windows, Unix, IBM AS/400, etc.) estarien en la mateixa illa. Hi haurien mitjans de comunicació "inter-illa", per tal de moure informació d'una illa a una altra.

Per exemple:

- En el cas de tenir diversos entorns que es volguessin comunicar amb una Intranet sobre Windows, podrien utilitzar el protocol HTTP [rfc1945] o HTTPS [rfc2818].

- Si tinguéssim uns usuaris treballant amb màquines Windows que necessitessin accedir a uns fitxers emmagatzemats en una illa Unix, ho podrien fer instal·lant-hi l'aplicació Samba [SB01].

- Si l'aplicació de gestió de la organització va sobre AS/400 i els usuaris estan en un altre entorn es poden utilitzar emuladors TN5250 [rfc1205] per utilitzar-la.

En aquest escenari tenim dos problemes:

- El primer és la poca integració de les eines disponibles d' un entorn respecte un altre.

  Per exemple, si volem accedir a una màquina Windows amb Terminal Services des d'un entorn Unix, tenim dues opcions:

  - afegint un nou protocol al Windows Terminal Services per permetre no només connexions RDP [MS12], sinó també ICA pel qual existeix client per un nombre molt més gran de plataformes, instal·lant Citrix Metaframe [CT01]

  - fer servir l'aplicació *opensource*, rdesktop [RD01] en les màquines Unix

  En ambdós casos s'ha d'instal·lar una aplicació amb els problemes que aquest fet implica (manteniment, seguretat, suport, etc.). En el cas del MetaFrame el cost és elevat a causa de les llicències.

- El segon problema tracta de les tuples d'usuari/contrasenya que han de tenir les persones que utilitzen les aplicacions. Cada illa té un domini d'autenticació, per tant cada persona que utilitzi l'aplicació ha de tenir un usuari independent en els diversos entorns. Normalment només les contrasenyes són diferents, però a vegades fins i tot es dóna el cas de tenir noms d'usuaris diferents, sobretot en fusions o adquisicions d'organitzacions, si el nom d'usuari en un entorn ja ha estat assignat a una altra persona.

  En aquest cas, l'usuari estarà descontent per haver de recordar diversos parells usuari/contrasenya per als diferents entorns. Però a més a més, segurament tindrem contrasenyes menys segures, per tant, més fàcils d'endevinar, o bé trobarem més *post-it*® en l'entorn de treball amb les diverses combinacions. Si les polítiques de seguretat de l'empresa no permeten aquestes "solucions" per part dels usuaris, probablement tindrem moltes més incidències a *Help Desk* per errors o oblits de contrasenya.

El nostre projecte no contempla el primer problema. No intenta trobar-li solució, ja que al mercat existeixen productes perfectament capaços de resoldre la majoria de les comunicacions "inter-illa". D'altre banda, hagués implicat tenir-nos que cenyir a un sol tipus de comunicació, i a un petit

nombre de protocols diferents. Haguéssim tingut problemes de temps i de recursos disponibles, tenint en compte, per exemple, que Citrix ha pogut crear el MetaFrame degut a que va llicenciar el codi font de Windows NT a Microsoft. És per això que hem optat per seguir un altre camí.

El projecte es centra en el segon problema. L'objectiu és facilitar a l'organització la feina, per que pugui implantar d'un sol domini d'autenticació, i que sigui lliure d'escollir quina tecnologia utilitzar, per exemple Ldap, Radius [rfc2138], usuaris d'AS/400, *Active Directory*, etc.

L'objectiu del projecte és basar-nos en l'entorn Windows NT/2000/XP/2003 a causa del temps i la informació. Per exemple, no teníem disponible cap AS/400, ni cap Apple, i perquè els sistemes Unix/Linux, ja tenen solucionat el problema amb els PAM.

Tot i així, creiem que tenim un mercat objectiu bastant ampli. Cada vegada més, la tendència es integrar els diversos entorns que té l'empresa. Aquesta, intenta aprofitar els avantatges que aporta cada entorn, per exemple:

- l'estabilitat del AS/400.

- la fiabilitat dels servidors TCP/IP i aplicacions *open source* competitives en Unix/Linux.

- facilitat d'ús i aplicacions d'ofimàtica i disseny en Windows.

Però a la vegada té por del què pot implicar el fet de tenir diversos entorns treballant junts, tant a nivell de disponibilitat com de compatibilitat o sobre costos en el manteniment. Per exemple, no és el mateix administrar un entorn amb 3.000 comptes d'usuari, que si tenim tres entorns diferents, amb 3.000 comptes cadascun, on a més estan en diverses màquines i diferents proveïdors d'autenticació, amb la qual cosa dificulta el diagnòstic del problema i el fet de solucionar-lo.

Les empreses han de complir determinades polítiques de seguretat informàtica, la primera que normalment s'implanta és la d'exigir complexitat i caducitat a les contrasenyes. Nosaltres hem analitzat diversos estudis que fan referència al cost que els hi suposa aquesta situació en un entorn amb diverses contrasenyes per persona. N'hem pogut extreure que l'usuari típic realitza 1'75 trucades a *Help Desk* al mes (Meta Group [MG01]), d'aquestes, un 30% són relacionades amb les contrasenyes (Gartner Group [GG01]). El cost per trucada és de $25 (Meta Group). Si implantem una solució que limiti el nombre de contrasenyes per usuari a una, reduïm les trucades relatives a aquest tema en un 85% (Datamonitor [DM01]).

Per tant, si tenim 3000 usuaris, tindrem un estalvi de costos en *Help Desk* de:

3.000 x 1'75 x 30% x $25 x 85% = $33.468'75/mes

Els estalvis anuals representarien $401.625 i ja que estem a Europa, 373.639€ (al canvi del 13/04/2003). Si parlem a nivell unitari, els estalvis serien de 124'5€/any només en *Help Desk.*

A més, hem que tenir en compte que mentre la incidència no està resolta, l'usuari afectat no pot treballar. Ja que estem tractant amb sistemes de validació d'usuaris, s'han de determinar uns procediments per restablir contrasenyes de forma segura, i impedir que hi hagi robatori d'identitat.

S'estima que des que s'obre la incidència, fins que *Help Desk* té prou informació per restablir la contrasenya de l'usuari, passen uns 7 minuts de mitja. Segons l'Institut Nacional d'Estadística [INE01], el guany mitjà per un treballador no obrer del sector serveis per hora es situava en 12'6 € l'últim trimestre de 2000 (no hem trobat públiques dades més recents), tenim que la pèrdua d'hores de treball a causa de d'incidències amb les contrasenyes seria de:

Cada 7 minuts tenen un cost de 1'47€

3.000 x 1'75 x 30% x 85% x 1'47€ = 1.967'96 €/mes o 23.615'52€/any o 7'8€/usuari/any.

En definitiva, si sumem el cost de *Help Desk* i el de la pèrdua de productivitat, tenim un estalvi de 132'3€/usuari/any.

Una solució d'aquest estil comença a tenir sentit quan hi ha un nombre elevat d'usuaris (si són pocs no es necessari un *Help Desk* o comprovació d'identitat) en un entorn heterogeni. Creiem que tenim la solució, estalviaria grans quantitats de diners, incrementaria la productivitat i possiblement faria que els usuaris no intentessin circumdar les polítiques de seguretat. Però les aplicacions informàtiques de l'organització han d'estar programades específicament per a ella, o bé embolcallar-les (*wrapper*) amb alguna altra que redirigeixi les funcions d'autenticació vers la PAM^{2}w, per tal que es pugui implantar correctament.

Veient aquest nínxol de mercat, també és una bona motivació per seguir endavant i provar de comercialitzar la plataforma. Per exemple en serveis de consultoria basats sobre aplicatius de codi lliure. De fet, els creadors de la *pGina* (*v. pGina p. 29*) acaben d'establir-se en una empresa on ajuden a d'altres organitzacions a implantar la seva solució.

En aquest anàlisi de viabilitat econòmica, només hem tingut en compte els estalvis que pot proporcionar a una empresa una solució semblant a aquesta. Hi ha estudis realitzats, per importants empreses d'analisi de mercat, que ens han permès determinar un cost a cada cas. Tot i això s'ha de tenir en compte les possibles oportunitats de negoci o baixada de costos que obriria una solució com aquesta en diversos escenaris.

Per exemple:

- **ASP (*Application Service Provider*):** Un ASP hauria de valorar quins avantatges li proporciona el fet de poder consolidar múltiples clients, amb necessitats diferents d'autenticació en un mateix servidor. Un possible cas, seria el d'oferir un servei de Microsoft *SQL Server* als seus clients, delegant tota la gestió, seguretat i disponibilitat, al ASP. Llavors l'ASP necessita *N* màquines en un clúster SQL per oferir una alta disponibilitat. S'ha de ser plenament conscient que aquestes maquines estaran infrautilitzades; però per SLAs (*Service Level Agreement*) s'ha compromès amb el seu client a oferir una disponibilitat, que sense una solució en clúster no pot garantir. Per tant, la solució que queda és passar el cost total a un sol client. En canvi en una solució basada en una Plataforma com la que ens ocupa, permetria tenir diverses instàncies del servidor SQL totes en clúster, fent servir les mateixes màquines, amb la qual cosa el cost d'infraestructura es pot repartir entre els diversos clients, oferint a l'ASP un producte més competitiu o bé amb més marge.

- **Organització Homogènia:** En el cas d'una organització on tots els sistemes siguin homogenis, segurament s'hauran trobat en la disjuntiva algun dia d'aprofitar els avantatges d'un altre sistema (per aplicacions disponibles, facilitat d'ús, cost, llicències, etc). Però per costos en la implantació o incompatibilitats amb la infraestructura actual, s'ha decidit no seguir el camí més òptim a priori. Amb la plataforma permetríem la implantació de sistemes *Microsoft Windows* en empreses on els seus sistemes fins ara eren incompatibles amb una solució *Windows* integrada dins el sistema d'informació de l'organització.

Hi hauria més exemples, però creiem que amb aquest dos es veu clarament. L'increment de negoci en un cas o la millora de l'eficiència en el segon poden superar clarament els estalvis per la gestió centralitzada d'usuaris. Però per reflexar-ho en euros s'hauria d'estudiar cas per cas, i potser en el de l'ASP, a posteriori de la implantació, veient com evoluciona el mercat.

Com autentica Windows

Els sistemes operatius Microsoft Windows NT/2000/XP/2003 poden autenticar els usuaris:

- segons una base de dades d'usuaris local a la màquina.

- a través de la xarxa amb el que es coneix com a Dominis.

En el cas de l'autenticació local, el sistema operatiu consulta en el registre [MS08] de Windows si l'usuari i la contrasenya són correctes. Els permisos de les branques del registre on hi ha la informació d'usuaris i contrasenyes (HKLM\Security\SAM (Security Account Manager)), per defecte només permeten la lectura i modificació dels valors a l'usuari SYSTEM, que es l'usuari que fa servir el propi Windows.

Per tal de permetre l'autenticació d'usuaris per aplicacions que no formen part del Sistema Operatiu, s'utilitza un mètode de delegació de privilegis on hi ha un procés corrent en espai d'usuari (*v. Glossari*), l'esmentat procés és el *Local Security Authority SubSystem* (LSASS), propietat de l'usuari **SYSTEM**. Aquest procés és el que s'encarrega de decidir si un combinació nom d'usuari/contrasenya són vàlides o no. Les aplicacions que volen saber si la informació que tenen sobre l'usuari correspon a un que estigui donat d'alta al sistema, li pregunten al LSASS mitjançant *Local Procedure Calls* (LPC), una mena de *Remote Procedure Call* (RPC) [rfc1050] adaptat a comunicar processos locals.

En el cas d'autenticar per domini, la comunicació es realitza del LSASS de la màquina que fa la petició al de la que proveeix l'autenticació, anomenada Controlador de Domini. Amb les diverses iteracions de la plataforma, s'han anat modificant tant els mètodes de comunicació com els llocs on guardar la informació de l'usuari.

NT 3.5x, 4.0:

La base de dades resideix en el registre, és la SAM. Només hi ha una màquina en tot el domini que hi pot escriure, el *Primary Domain Controller* (PDC). La resta de controladors de domini tenen la base de dades en mode de només lectura, els *Backup Domain Controller* (BDC). Periòdicament es van sincronitzant amb el PDC.

Per exemple, un canvi de contrasenya només es pot fer sobre la base de dades del PDC, tot i que la validació d'usuari es pot fer en qualsevol BDC, normalment el més proper a la màquina que fa la petició. Si parlem de seguretat, la comunicació entre les diverses màquines del domini es realitza mitjançant els protocols NTLM i NTLMv2 (depenent de nivell de Service Pack). Per més informació sobre el protocol veure [MS04].

2000/2003:

En aquestes versions, la base de dades d'usuaris del domini es diu *Active Directory.* En realitat, no és més que una Base de Dades optimitzada per a dades semi-estructurades i basada en arbres B [CC02]. S'anomena *Extensible Storage Engine* (ESE) i proporciona interfícies de consulta Ldap i Kerberos. Aquí, la base de dades d'usuaris és *multimaster*, amb la qual cosa es poden realitzar modificacions en més d'un servidor i replicar-se entre ells. És per això que aquí perdem

el concepte de PDC i BDC per passar a ser tots *Domain Controller* (DC). De totes maneres, per millorar l'escalabilitat hi ha el concepte de **sèus**, on estan agrupats tots els DC que tenen bona connectivitat entre ells, per tal de minimitzar el trànsit sobre enllaços costosos (*FrameRelays, VPNs, XDSIs, etc*). A nivell de comunicació, la seguretat funciona amb Kerberos [MS05], on cada DC és un *Key Distribution Center* (KDC). Explicat d'un altre manera, és un dispensador de tiquets, així si un procés posseeix un tiquet vol dir que algun KDC ha acceptat la seva validació. Llavors al presentar-lo a qualsevol altre procés, aquest, ja sap que representa a un usuari validat. Per més informació sobre Kerberos veure [MIT01].

Com s'acostuma a gestionar els usuaris en Windows

L'objectiu d'aquesta secció és descriure com diversos aplicatius Windows gestionen usuaris. Els podríem categoritzar en quatre grans grups:

- **Nadius Windows**: Es basen en l'autenticació que proporciona Windows. Això té avantatges i inconvenients. El principal avantatge és que només s'ha d'administrar un conjunt d'usuaris i ens podem beneficiar de l'entorn de domini. Els principals inconvenients són la gestió que l'aplicació faci de la informació d'usuari i com es transmeti aquesta. Per exemple, en el cas d'un FTP amb autenticació Windows, tenim com a eines de gestió, el registre d'accions i seguretat del propi sistema operatiu. En canvi si algú captura paquets de la xarxa, al ser un protocol no encriptat, poden comprometre tot l'entorn i no només el servidor FTP. Un desavantatge més prosaic és que per cada usuari de Windows necessitem una llicència i a vegades el servei a oferir per l'aplicació no justifica el cost.
- **Gestió Pròpia**: El propi programa gestiona la base de dades d'usuaris. En aquest escenari també tenim pros i contres. El principal avantatge és que si algú compromet alguna compta, estarà restringit als serveis d'aquella aplicació. En canvi, es complica la gestió d'usuaris ja que s'han de crear eines i processos per tractar amb el proveïdor d'autenticació específic del servidor, a més d'haver de formar a *Help Desk* en la nova aplicació.
- **Ús d'altres proveïdors**: En aquest cas, l'aplicació aprofita proveïdors d'autenticació externs. Segurament ja no haurem d'utilitzar eines especifiques de l'aplicació, sinó eines del proveïdor i el cost de formació o desenvolupament queda repartit entre les diverses aplicacions. El problema arriba quan una aplicació ens demana que implantem un proveïdor d'autenticació diferent al que ja tenim. Llavors estem davant de la disjuntiva d'implantar el nou proveïdor o, buscar una altra aplicació encara que potser sigui menys idònia per la tasca, que la que estàvem avaluant.
- **Híbrids**: Són aquelles aplicacions que no entrarien a cap de les altres categories ja que podem configurar el tipus de proveïdor d'autenticació que farem servir. Normalment tenim un sistema de gestió pròpia i un altre d'externa que pot ser Windows, Ldap o Radius [RFC2865], per citar els més comuns.

Com podem veure, cap de les categories ens proporciona un entorn flexible de gestió d'usuaris. L'administrador de sistemes mai pot decidir quin sistema d'autenticació farà servir un determinada aplicació, com a molt se li dóna un petit ventall d'opcions entre les quals ha d'escollir-ne una. La finalitat d'aquest projecte és, que la tria del proveïdor d'autenticació que ha de fer servir una aplicació, la pugui fer l'organització que l'utilitzarà i no els desenvolupadors de l'aplicació.

Planificació del Projecte

El projecte ha estat dividit en tretze apartats, dos dels quals: el desenvolupament del SDK i els exemples, tenen dos subapartats cadascun.

Les diverses tasques són:

Definició d'Objectius: Destinat a limitar fins on volem arribar i quins dels múltiples objectius considerem més importants i per tant realitzarem.

Estudi de Mercat: Estudiar quines són les solucions existents. Què podem fer per tal de desmarcar-nos i aconseguir una franja de mercat. Veure pros i contres de cada enfocament per poder agafar el millor de tots i corregir errors.

Elecció de la Solució: Després de veure que hi ha al mercat, decidir que hem de fer i com fer-ho.

Anàlisi de Requeriments: Quins requeriments ens imposa la pròpia solució escollida i el mercat objectiu.

Especificació: Definir com ha d'interactuar la solució amb l'entorn.

Disseny: Definir l'arquitectura de la solució.

Escollir eines d'implementació: Un cop sabem el que volem fer, quines són les eines més òptimes per dur-ho a terme?

Implementació PAM^{2}w: Implementar el nucli de la plataforma.

Documentació PAM^{2}w: Documentar per a què serveix cada funció, eleccions, etc. La documentació es realitza paral·lelament a la implementació.

Control de Qualitat: S'allarga des de l'especificació fins que ja hem acabat tots els mòduls. Va verificant que el producte s'adiu amb els requeriments i que totes les parts treballin conjunta i correctament.

Desenvolupament del SDK: Proporciona les eines per augmentar l'abast de la plataforma a nous proveïdors d'autenticació i a noves aplicacions. Està dividida en dues tasques:

Disseny: Arquitectura del SDK. Com hem de dividir les funcionalitats.

Implementació i Documentació: Implementar i documentar l'SDK. La documentació és molt important, ja que en principi ho utilitzarà gent que no tenen perquè ser experts en la PAM^{2}w

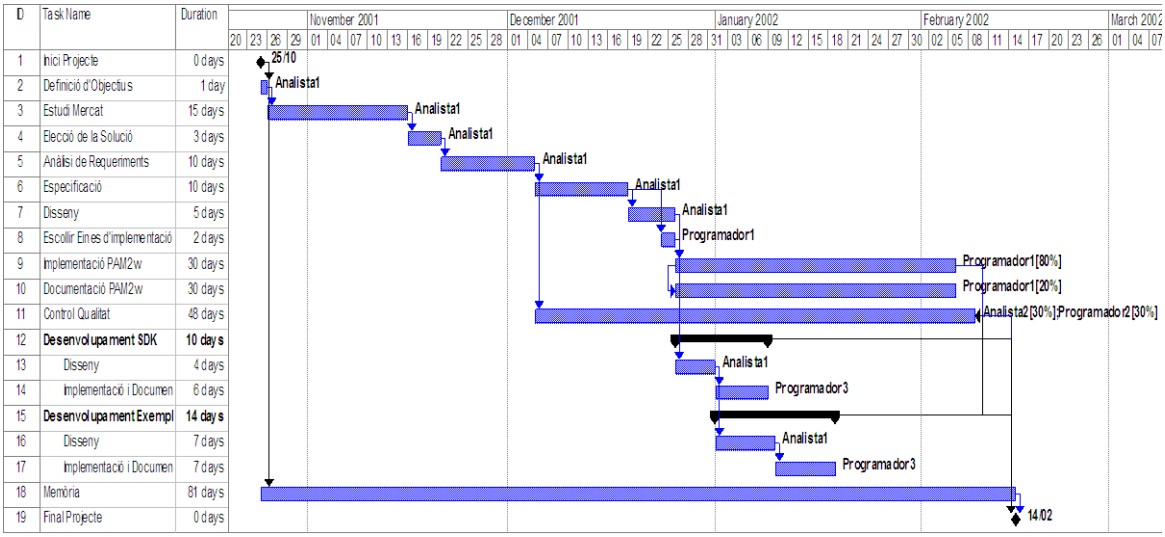
Desenvolupament dels Exemples: Exposicions d'exemples, tant de mòduls com d'aplicacions. Per poder fer el control de qualitat i sobretot, per poder donar un entorn pedagògic als desenvolupadors. Està dividida en dues tasques:

Disseny: Arquitectura dels exemples, com hem de dividir les funcionalitats.

Implementació i Documentació: Implementar i documentar els exemples. Aquí, també, la documentació és molt important, ja que la idea és que el puguin fer servir de material de referència.

Memòria: L'evolució de tota la documentació del projecte. Es va realitzant en paral·lel amb el projecte i ens reservem quatre dies per formateig i revisió d'errors.

En les planificacions del document s'ha utilitzat un preu de 30€ per hora d'analista i de 15€ per hora de programador. La realització de la Memòria no ha estat inclosa en el càlcul dels costos ja que és una tasca que forma part de la resta de les feines.



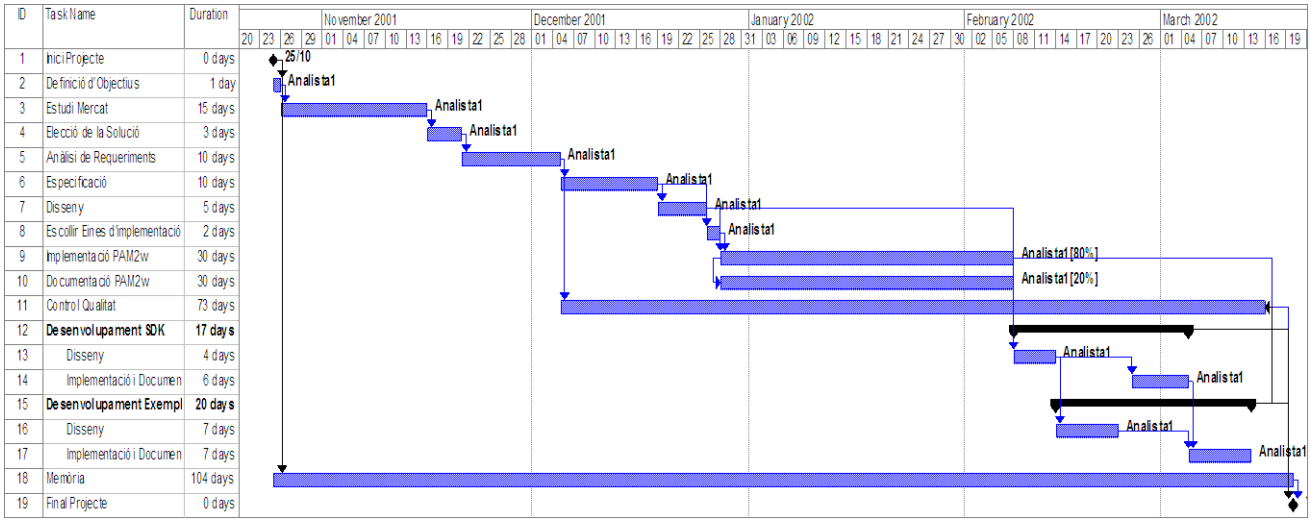
Malgrat suposar que tenim tots els recursos humans i materials, sempre tindrem un mínim de duració imposat. Hi han tasques que depenen de la finalització de les anteriors per poder-les començar.

Per assolir el Projecte en el temps mínim de realització, necessitem disposar dels recursos següents:

- dos analistes
- tres programadors
- un personal independent dels anteriors per realitzar el control de qualitat (ja que seria força probable que no s'adonés d'errors d'interpretació dels requeriments, especificació o disseny).

Aquesta planificació indica que el projecte duraria 81 dies i tindria un cost de 19.118,40€, repartits en els diversos conceptes i en costos de recursos humans:

Definició d'Objectius	240,00 €
Estudi Mercat	3.600,00 €
Tria de la Solució	720,00 €
Anàlisi de Requeriments	2.400,00 €
Especificació	2.400,00 €
Disseny	1.200,00 €
Escollir Eines d'implementació	240,00 €
Implementació PAM^{2}w	2.880,00 €
Documentació PAM^{2}w	720,00 €
Control Qualitat	518,40 €
Desenvolupament SDK	1.680,00 €
Desenvolupament Exemples	2.520,00 €



Si només tenim un recurs humà, es a dir el projectista, la feina s'allarga en el temps, ja que tenim una restricció més forta que la de les dependències entre tasques.

Com que hem reduït el nombre d'hores diàries de dedicació al projecte, aquest s'allarga fins el 19 de Març de l'any 2002.

Ja que només tenim una persona, no farem distincions entre analista i programador. En principi un analista sap fer la tasca de programador, per tant considerem que la persona que tenim disponible és un analista. El cost de personal també puja pel fet de tenir moltes més hores d'analista.

Per tant, ara, el cost del projecte puja fins als 25.051,20 € que desglossat queda:

Definició d'Objectius	240,00 €
Estudi Mercat	3.600,00 €
Elecció de la Solució	720,00 €
Anàlisi de Requeriments	2.400,00 €
Especificació	2.400,00 €
Disseny	1.200,00 €
Escollir Eines d'implementació	480,00 €
Implementació PAM2w	5.760,00 €
Documentació PAM2w	1.440,00 €
Control Qualitat	1.051,20 €
Desenvolupament SDK	2.400,00 €
Desenvolupament Exemples	3.360,00 €

Malgrat tenir un cost de personal elevat, és molt menor que l'estalvi potencial que pot tenir un client. Anteriorment hem vist que en un entorn heterogeni com el de la FIB amb uns 3.000 usuaris, s'obtenen uns estalvis de 373.639€/any. És per això que hem decidit tirar endavant el projecte, tot i no saber del cert el capital que haurem d'invertir en llicències i màquines per portar-lo a terme, estimem que serà molt inferior a 20.000€ i per tant encara tenim un retorn de la inversió atractiu. A més s'ha de tenir en compte que es pot aprofitar la mateixa base de codi per a d'altres organitzacions, millorant encara més el rati inversió/ingressos.

ANÀLISI DEL MERCAT

En aquest capítol comentarem les diverses opcions disponibles al mercat, tant per a Microsoft Windows com per d'altres entorns.

NtLdap

Aquesta solució no està realment al mercat però, està disponible en entorns reduïts. **NtLdap** fou el treball que vaig desenvolupar dirigit per Leandro Navarro on s'intentava resoldre l'accés interactiu al sistema.

NtLdap funciona de la següent manera (v. Fig.1):

L'usuari escriu el nom d'usuari i la contrasenya en el diàleg que se li mostra.

NtLdap intenta validar-lo en el proveïdor d'autenticació de tipus Ldap que l'administrador hagi configurat per la màquina. Si la validació és acceptada, **NtLdap** intenta validar localment en el Windows, ja sigui mitjançant la Base de Dades local d'usuaris (SAM) o bé a través del Domini NT de la màquina. Si l'accepta localment, es concedeix l'accés interactiu a l'usuari.

Si hi ha hagut un error en la primera validació, vol dir que el nom d'usuari i la contrasenya no són reconegudes pel proveïdor d'autenticació configurat, amb la qual cosa retorna l'error immediatament.

Si l'error es troba en la segona validació, ens trobem amb un usuari vàlid pel sistema però no reconegut per la màquina local. Això pot ser donat perquè:

- un mateix usuari te contrasenyes diferents en el Ldap i la màquina local.

- l'usuari no està donat d'alta en la màquina local.

Per solucionar el problema, primer s'intenta canviar la contrasenya local per sincronitzar-les. Si no n'hi ha prou, s'intenta donar d'alta l'usuari en la màquina local. Si de totes maneres no es pot donar d'alta el nou usuari, això significa que hi ha hagut un error greu i l'usuari l'hauria de notificar a l'administrador del sistema perquè actuï.

NtLdap és un substitut de la Graphical Identification and Authentication *(GInA*), per tant, implementa la interacció amb l'usuari. Per tal d'ampliar la flexibilitat i facilitat de modificació es realitzà la interfície amb HTML amb l'opció de baixar-se'l per HTTP o FTP en el moment de la presentació. Un cop ja hem aconseguit l'usuari i la contrasenya el procés de decisió és el següent:

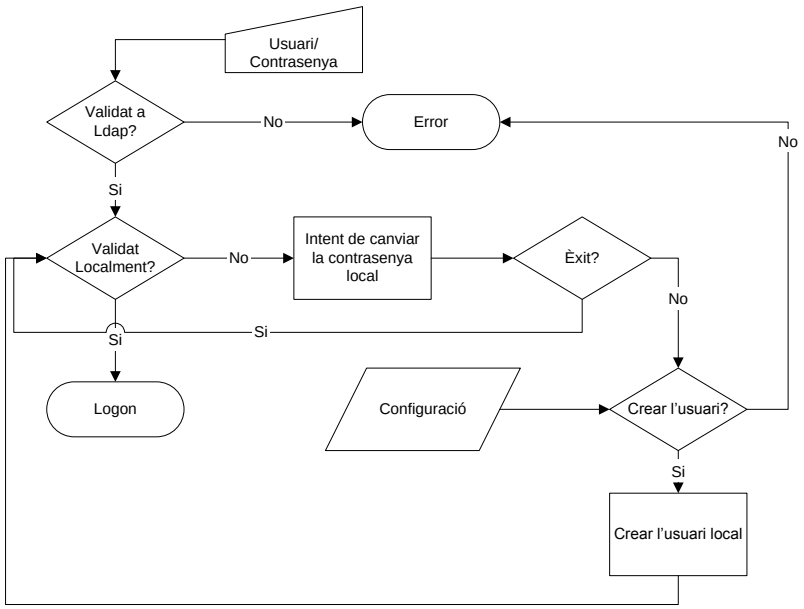


Figura 1

El que aconsegueix **NtLdap,** és que les estacions de treball o servidors amb servei de terminals basats en Microsoft Windows NT 4.0/2000/XP/2003, puguin sincronitzar-se automàticament amb un servidor de directori mitjançant el protocol LDAP [RFC4510] amb SSL [NS01].

Les mancances de **NtLdap** són diverses:

- Només tenim un altre proveïdor d'autenticació, Ldap. Que tot i ser un dels més comuns en entorns heterogenis no soluciona totes les problemàtiques. Per tant manca flexibilitat.
- Només soluciona el problema de l'accés interactiu. Un cop l'usuari ha entrat, tot funciona com si fos un usuari local normal, de fet és així, ja que si la compta no estava donada d'alta, se n'ha donat una d'alta al mateix moment d'entrar (depenent de la configuració).
- Ja que només emula parcialment un entorn de domini (les comptes sempre són locals, creant-se i eliminant-se amb l'inici i final de sessió d'usuari), no ens assegura que la propera vegada que l'usuari torni a validar-se pugui accedir els fitxers dels quals era propietari i només ell en tenia permisos. Això és degut a que els permisos de fitxers en les particions NTFS (NT FileSystem) estan assignats al SID (*Security Identifier*) de la compta i aquest s'assigna dinàmicament durant la creació de la mateixa, per tant si entre la creació i accés del fitxer s'ha borrat la compta i tornat a crear, l'usuari no se n'adonarà però no podrà accedir als fitxers. Per això es pot configurar si les comptes es donen de baixa quan finalitza la sessió o bé es mantenen actives per tornar a aprofitar el SID la següent vegada que l'usuari accedeixi.

pGina

pGina [PG01] és un projecte de la Pacific Lutheran University (http://www.plu.edu) distribuït sota la llicència GPL [OSI01] i dirigit per Michael Wright i Nathan Yocom.

Com el **NtLdap,** pGina també és un substitut de la *GInA* dels sistemes operatius Microsoft Windows NT 4.0/2000/XP. La diferència més important entre ambdós reemplaçants és que **pGina** permet treballar amb *plugins,* amb la qual cosa no ens limita a Ldap sinó que podem escollir diversos proveïdors d'autenticació o bé desenvolupar el nostre propi.

Principals inconvenients:

- Només resol l'accés interactiu. Està força limitat des del punt de vista de la interfície d'usuari, al haver de modificar codi per adaptar-la a les necessitats de l'entorn.
- Quan un usuari s'ha validat en el proveïdor escollit per l'administrador del sistema, es crea una compta local per l'abans esmentat usuari. A diferència de **NtLdap**, es clona una compta local i per tant tots els clons de la mateixa compta formaran part del mateix grup.
- Quan l'usuari tanca la sessió ens podem trobar amb els mateixos problemes que si estiguéssim treballant amb **NTLDAP**. (v. Pàg 27)
- De tots els mòduls que tenen previstos els creadors de pGina, només són funcionals el de **Ldap** i un anomenat PAM. Aquest últim permet comunicar-se amb un servidor Linux on prèviament s'han configurat els PAM i la part servidora del *plugin*. Per tant tot i permetre *plugins* no s'estan potenciant excessivament en les primeres versions. Actualment i gracies a que han obtingut força tracció en el món Opensource, disposen d'uns 15 *plugins* per a diversos proveïdors d'autenticació.

Novell SecureLogin

És un producte de la marca Novell [NO01]. Té com a objectiu evitar que l'usuari hagi de veure's obligat a recordar múltiples parelles nom d'usuari/contrasenya per accedir a les diverses aplicacions de que disposi una organització.

L'enfocament que ha donat Novell al seu producte permet que només s'hagi d'instal·lar el SecureLogin a l'estació de treball. Tenim com a possibles proveïdors d'autenticació el **Novell eDirectory** [NO02], **Microsoft ActiveDirectory** [MS07] o bé fent servir l'autenticació local vers una base de dades local d'usuaris pròpia del SecureLogin.

Un cop un usuari s'ha validat localment amb el mòdul substitut de la *GInA* del **SecureLogin**, aquest últim busca en el proveïdor d'autenticació configurat, quin nom d'usuari i contrasenya de domini NT posseeix l'usuari que acaba d'entrar, i el valida en el domini. Quan l'usuari executi alguna aplicació que demani nom d'usuari i contrasenya, el **SecureLogin** actuarà de tal manera que l'usuari no hagi d'intervenir per omplir les dades. Capturarà la petició, identificarà quina aplicació està fent la sol·licitud i buscarà a la base de dades quina informació corresponen a aquella aplicació.

Els principals avantatges de la solució són:

- Les aplicacions no han d'estar dissenyades per el **SecureLogin**.
- Les parelles nom d'usuari/contrasenya no han de ser totes iguals per un usuari com passa amb la sincronització. D'aquesta manera si una compta queda compromesa per disseny de protocol o mala gestió, només queda afectada aquesta, no pas la resta d'entorns on l'usuari té comptes.

Com a desavantatges tindríem:

- Els noms d'usuari i contrasenyes estan en un fitxer local, per tant és factible suposar que si es perd el control sobre la màquina, alguns o tots poden quedar compromesos. Per exemple, si s'extreu el disc dur, on hi ha el fitxer de noms d'usuari i contrasenyes, i s'instal·la en un altre màquina on hi hagin eines per extreure la informació, seran inútils les restriccions a l'accés al contingut del fitxer.
- Si es dones el cas que hi haguessin comptes compromeses, la única solució seria la de canviar totes les contrasenyes a aquestes. A més a més al no proporcionar un entorn únic de gestió de comptes, s'hauria de canviar una a una les contrasenyes a cada proveïdor on l'usuari tingues comptes.
- Problema amb possibles Trojans (*v. Glossari*) fent-se passar per aplicacions on el **SecureLogin** introduirà la contrasenya i per tant comprometent una compta.
- Potencialment hi haurà aplicacions que no seran compatibles amb el **SecureLogin,** ja sigui perquè no podrà detectar el diàleg demanant el nom d'usuari i contrasenya; o bé perquè l'aplicació utilitza tecnologies de biometria, on l'usuari haurà d'actuar i introduir alguna mena de codi.

PAM (Pluggable Authentication Modules)

Aquesta solució és la més estesa en entorns Unix/Linux, tant és així que la majoria l'usen com a sistema d'autenticació per defecte.

Els primers en veure la utilitat de tenir un punt entremig en l'autenticació d'usuaris, van ser els membres de l'equip de desenvolupament del SunOS, on ja a l'any 1996 van publicar els primers documents i presentacions sobre els PAM [SM02].

Havien de fer front a un problema. Aquest era que la majoria de programes d'accés remot del seu sistema operatiu, tenien implementats tots els proveïdors d'autenticació a utilitzar. Això provocava que l'administrador havia de controlar mitjançant fitxers de configuració diferents quin proveïdor feia servir cada programa. Tenint en compte q cada canvi s'havia d'implementar a cada un dels programes, el desenvolupament de nous proveïdors d'autenticació o arreglar defectes en els existents comportava una gran sobre cost al ser cada vegada més nombrosos els programes i proveïdors d'autenticació a suportar per SunOS.

La solució que es proposà fou la de separar el proveïdor d'autenticació de l'aplicació, així un canvi en el protocol d'autenticació o la implantació d'un de nou no implicava refer-ho tot. Per tal d'aconseguir aquest objectiu, es creà l'arquitectura dels PAM, on hi havia una petita peça de *software* que feia de mitjancera entre l'aplicació i el proveïdor d'autenticació. Així només s'havia de programar l'aplicació per a un proveïdor d'autenticació, els PAM. Les actualitzacions o novetats també s'havien de programar una sola vegada. La configuració de quin proveïdor d'autenticació feia servir cada programa estava centralitzada en un sol fitxer, on s'indica per a cada programa quin o quins mòduls ha de fer servir.

La versió actual de PAM no només proporciona mòduls d'autenticació, sinó que n'ofereix més, concretament 4 tipus:

Nom	Referència	Descripció
Autenticació	auth	Aquest tipus de mòduls comproven que l'usuari és qui diu que és. Ho fan mitjançant algun proveïdor d'autenticació, a més a més pot assignar l'usuari a un grup o li pot donar d'altres privilegis.
Comptes	account	Aquests permeten accés mitjançant altres consideracions que la pròpia autenticació, com pot ser només deixar entrar un usuari unes hores del dia, o bé, depenent des d'on s'està accedint a l'aplicació.
Sessió	session	Realitza les tasques necessàries (abans o després segons estan establertes) per poder donar el servei: afegir l'entrada en el registre, preparar l'entorn per l'usuari, etc...
Contrasenya	password	Gestió de les contrasenyes de cada proveïdor.

A més hi ha dues formes de guardar la configuració, mitjançant un fitxer comú per a totes les aplicacions amb aquest format:

```
#nom-programa    module-type    control-flag    module-path    arguments
#
# default; deny access
#
login    auth      required       /usr/lib/security/pam_deny.so
login    account   required       /usr/lib/security/pam_deny.so
login    password  required       /usr/lib/security/pam_deny.so
login    session   required       /usr/lib/security/pam_deny.so
```

O bé amb diversos fitxers amb el nom de programa, del mateix estil que l'anterior exemple però sense el camp *nom-programa* ja que ja esta indicat en com a nom de fitxer.

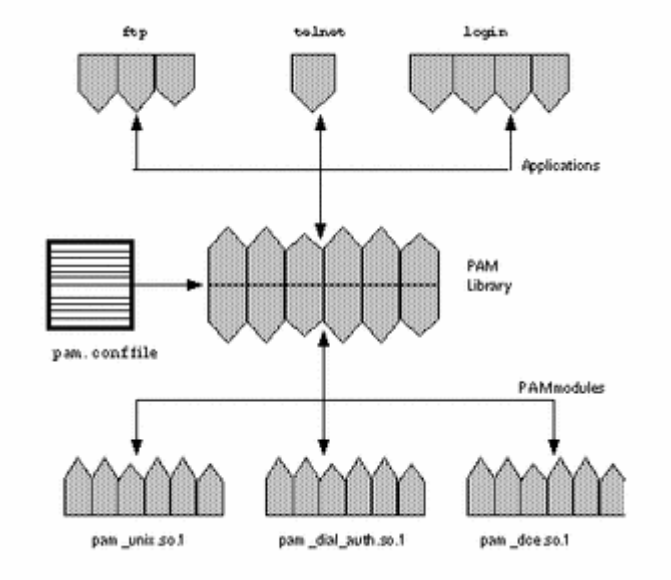


Figura 2

ESTRATÈGIES D'IMPLEMENTACIÓ

Aquí s'expliquen les diverses opcions que teníem per tal de proveir un entorn flexible tant per a desenvolupadors com per a administradors de sistemes.

Security Packages

Els *Security Packages* són la implementació en *software* d'un protocol de seguretat. Els *Security Packages* poden estar continguts dins d'un **SSP** o bé d'un **SSP/AP**. Els primers són els *Security Support Providers* que implementen el SSPI (SSP Interface) [MS09] proporcionant connexions autenticades, integritat i encriptació de missatges, mentre que els **SSP/AP** (SSP/ Authentication Package) a més de tot això són responsables d'analitzar si l'usuari pot accedir al sistema o no, d'establir una nova sessió i crear un identificador únic de la mateixa, i passar informació necessària a la LSA (*Local Security Authority*) per al *Token* de l'usuari.

Microsoft recomana utilitzar els SSP i SSP/AP per implementar nous mitjans d'autenticació i integrar-se en tota la infraestructura d'autenticació i autorització ja implantada per Microsoft i els seus socis. Tot i ser el mitjà recomanat oficialment, entenem que no és la millor opció per assolir els objectius que ens hem marcat. Desenvolupar una aplicació que treballi directament amb els SSPI és complexa ja que aporten molta més funcionalitat de la requerida i a la vegada no estan gaire documentats, a més és l'aplicació la que tria quin sistema d'autenticació ha de fer servir, reduint la llibertat dels administradors a decidir quina aplicació fa servir quina autenticació, o si per d'altra banda les aplicacions permeten definir-ho, incrementem la complexitat al guardar cada aplicació al dipòsit de paràmeters més convenient per a ella.

Durant l'estudi de les opcions vam veure que per a un desenvolupador sense experiència prèvia en els SSPI, li era molt més complex desenvolupar per a un SSPI que no pas implementar el sistema d'autenticació que més s'escaigui a les seves necessitats, així que el més probable és que no s'utilitzi tot el que ens agradaria.

Per últim no resolen tampoc l'accés interactiu al sistema, ja que ha de ser la pròpia *GInA* de Microsoft la que esculli el nostre proveïdor d'autenticació i és un dels grans problemes que es presenten en els entorns on tenim experiència.

Mòduls Monolítics

Aquesta opció consistiria en que cada mòdul implementés tota la comunicació amb el proveïdor d'autenticació i a més implementés tota la interfície vers l'aplicació que l'usa.

Escollir els **Mòduls Monolítics** per a implementar la solució implicaria que els desenvolupadors de mòduls haguessin de programar més codi, amb els possibles problemes que podria causar, més codi, més *bugs*, i és probable que el fet d'haver de programar més codi freni el desenvolupament *in-house* per part d'algun administrador de sistemes.

A més per tal d'aconseguir implantar la solució en aplicacions que formin part de la línia de negoci, hem de poder assegurar compatibilitat comprovada entre mòduls i aplicació. Si cada mòdul parla directament amb l'aplicació, les proves s'han de fer pel producte cartesià entre mòduls i aplicacions.

Un altre problema és la falta de flexibilitat i coherència al configurar quins mòduls ha de fer servir cada aplicació ja que, és probable que cada aplicació imposi el lloc de la configuració on es defineix quin mòdul fa servir i a més, és possible que el nom del mòdul, camí d'accés o d'altres paràmetres estiguin definits dins del codi o siguin menys accessibles del que seria de desitjar, al definir cada aplicació les seves eines i mitjans de configuració.

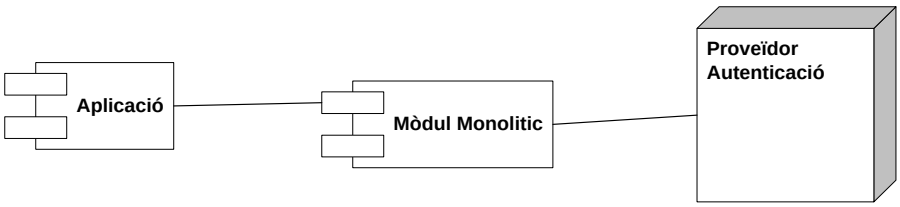


Figura 3

GInA amb Authentication Packages

Aquesta és la solució de Microsoft per al *logon* interactiu. L'usuari interacciona amb una *GInA* personalitzada (la de Microsoft **sempre** valida amb l'autenticació de Microsoft), aquesta usa la funció *LsaLogonUser* de la *Local Security Authority* i aquesta farà servir l'*Authentication Package* indicat a la crida de la *LsaLogonUser*, el procés és (v. fig 4):

1. El procés **Winlogon** és engegat pel Sistema Operatiu al iniciar-se.
2. El **Winlogon** després d'haver inicialitzat les seves dades, carrega al seu espai de memòria la **GInA** configurada i aquesta mostra el diàleg a l'usuari.
3. L'usuari prem la *SAS* (*Secure Attention Sequence*, típicament *Control+Alt+Supr*)
4. El controlador de teclat, després de detectar aquesta seqüència avisa al **Winlogon** que al seu temps avisa a la **GInA** per tal que mostri un diàleg a l'usuari per tal que aquest introdueixi les dades necessàries per autenticar-lo.
5. Un cop l'usuari ha entrat les dades, la **GInA** cridarà a la funció *LsaLogonUser* amb la informació que li ha passat l'usuari i indicant al **LSA** (*Local Security Authority*) quin **Authentication Package** ha de fer servir.

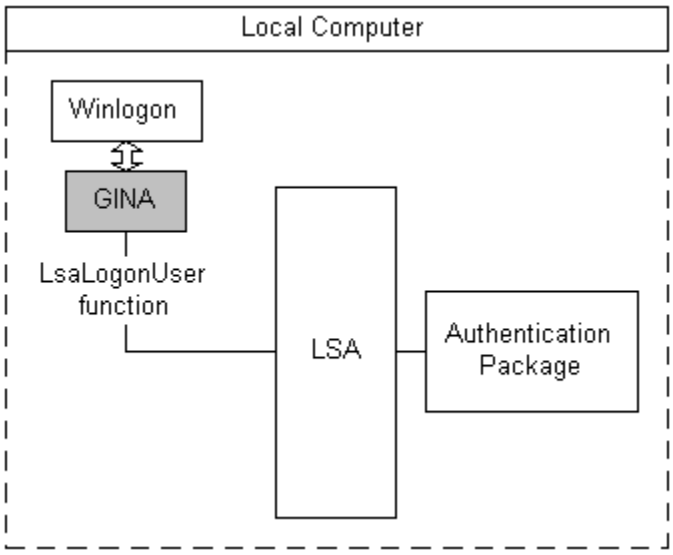


Figura 5

PAM

Com hem comentat en el capítol anterior al fer un estudi de les solucions ja existents en el mercat, els **PAM** (*Pluggable Authentication Modules*) al tenir una arquitectura molt modular i interfícies senzilles, proporcionen una gran flexibilitat tant per als desenvolupadors de mòduls d'autenticació com als d'aplicacions, a més de permetre un ràpid prototipatge i desenvolupament.

Basar la nostra solució en una arquitectura similar a la dels **PAM** ens proporcionaria els avantatges de que ja gaudeixen els entorns Unix/Linux a l'entorn Windows.

El principal avantatge de tenir un entorn d'autenticació flexible i fàcilment personalitzable, és que, al no haver de dependre d'un sol entorn d'autenticació els administradors o organitzacions poden triar la solució que més s'adequa a les seves necessitats i així reduir costos per ineficiències en la gestió de la infraestructura IT.

Migrant els **PAM** a Windows també aconseguim que sigui relativament fàcil la migració de mòduls existents per a **PAM** en d'altres entorns, segons el dipòsit de codi de Kernel.org [KO01], hi ha uns 68 mòduls PAM diferents, alguns de codi font obert (la majoria) i d'altres de tancat i van des de mòduls que validen amb fitxers a text fins a reconeixedors de veu.

Un exemple podria ser, aprofitar servidors Linux amb Samba, OpenLdap i servidor de Correu, tot *opensource* per baixar els costos de llicències, però que els usuaris continuïn usant com a eina les aplicacions per entorn Windows, amb això aconseguim que els usuaris no han variat la forma de treballar però el cost en llicències ha baixat de forma brusca (no n'hi ha de part servidora) i potser s'ha incrementat la disponibilitat del servei, a part de segurament baixar els costos operacionals del departament tècnic a l'estandarditzar sobre el mateix entorn i per tant aprofitar el coneixement existent en l'organització en Linux.

Per un altre, potser més comú, seria el d'una organització amb història, on es començà amb Mainframes o Miniordinadors (gairebé sempre IBM) i per necessitats del mercat ara ens obliguem a tenir un entorn d'ofimàtica basat en Windows, Office i navegador Internet. L'opció més típica és crear un nou entorn amb duplicitat d'usuaris ja que només ho necessiten uns quants directius i és gestionable, el problema ve quan al passar el temps la majoria de personal treballa amb els dos entorns i aquests no estan integrats, incrementant la probabilitat d'incidents en la gestió de credencials d'accés a les aplicacions.

Elecció

L'opció que hem triat per a desenvolupar el projecte ha estat la de migrar els PAM de l'entorn Unix/Linux a l'entorn Windows, així podem aprofitar l'experiència d'administradors en la solució i exemples de codi ja existents.

El principal problema que suposem ens trobarem és la falta d'experiència que tenim al programar amb càrrega sota demanda de mòduls en Windows, que esperem superar mitjançant la documentació existent i possible ajuda a través de fòrums, grups de noticies o de la pròpia Microsoft amb la qual no hem tingut cap problema en el passat.

L'altre problema que detectem és com fer l'API prou flexible i senzill per assolir els objectius marcats sense allargar massa la part de disseny i que rebi tota la planificació del projecte, al ser un camí crític pel projecte.

ANÀLISI DE REQUERIMENTS

En aquest capítol estudiarem els requeriments que ens venen imposats per els altres desenvolupadors (de mòduls i aplicacions), i per part dels administradors de sistemes que són els que hauran d'implantar les solucions desenvolupades amb la PAM^{2}w.

De negoci pels desenvolupadors de la PAM²w

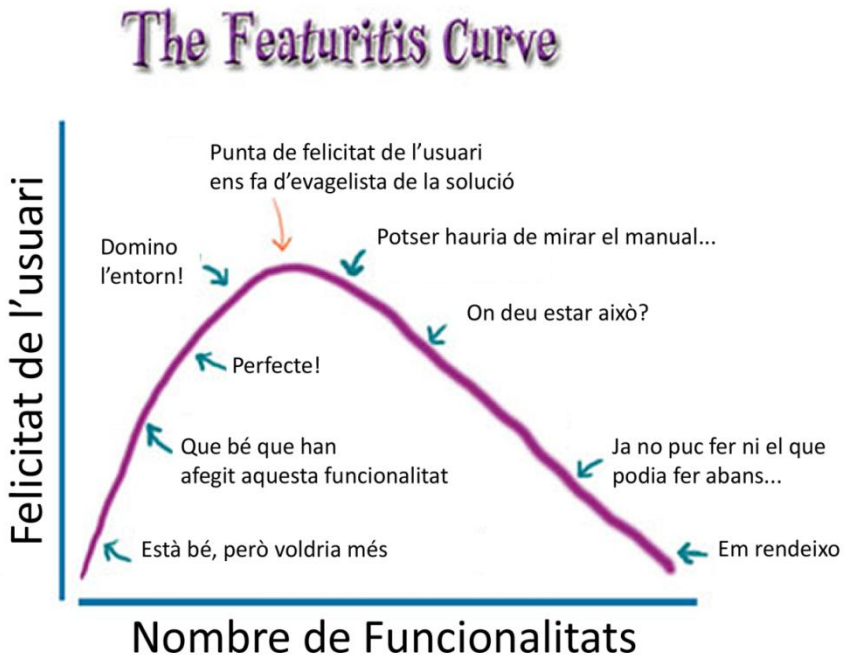
Estem desenvolupant una eina que entenem que disposa d'un mercat objectiu força ampli, i per tant podria ser font de negoci, ja sigui mitjançant models de llicenciament, ja sigui oferint lliurament la solució sota alguna llicencia de codi lliure i posicionar-nos com a millor suport per la implantació i operació d'entorns usuaris de la Plataforma.

Independentment del model de negoci escollit o fins i tot de la possibilitat d'explotar el treball realitzat de forma comercial, entenem que qualsevol projecte ha d'oferir uns avantatges suficients per tal que la seva implantació resulti rendible pels actors que hagin invertit en ella.

Per tant, creiem que la millor forma de complir el màxim de necessitats i per tant fer el màxim d'atractiva la nostra oferta, passa per aconseguir una massa crítica suficient, i la millor forma de realitzar-ho mantenint un pressupost de màrqueting ajustat, passa per fer que els propis usuaris de la Plataforma en parlin bé i la prescriguin a col·legues seus.

Una bona forma d'arribar a l'objectiu és trobar un bon equilibri entre complexitat de les interfícies i funcionalitat obtinguda per l'usuari, que es podria resumir en un gràfic on a l'eix de les abscisses tenim el nombre de funcionalitats, número de crides, paràmetres, capacitat per adaptar la plataforma a la més mínima necessitat i a les ordenades tenim la felicitat de la persona que treballa amb la interfície X. L'objectiu és aproximar-nos al punt on la majoria de possibles desenvolupadors és sentin còmodes adaptant les seves aplicacions a la PAM²w, de tal forma que obtindríem el millor màrqueting possible, que una persona del nostre objectiu de mercat recomani la solució a un col·lega seu que també pertany al mateix públic objectiu.

Els punt marcats en el gràfic fan referència al sentiment subjectiu d'un usuari X vers la PAM²w



Font: Kathy Sierra - http://headrush.typepad.com/creating_passionate_users/2005/06/featuritis_vs_t.html

Desenvolupadors d'aplicacions

Els desenvolupadors d'aplicacions que controlen usuaris demanden un entorn flexible d'autenticació. Per exemple totes les aplicacions Client/Servidor haurien de controlar els accessos. Vist que existeix la necessitat, l'objectiu és ampliar el mercat del producte. El fet de desenvolupar aplicacions que utilitzin la Plataforma, ha de comportar una rellevant implantació en els seus clients per tal que sigui rendible. La interfície de programació ha de ser de fàcil ús i flexible per evitar uns importants increment de les despeses de desenvolupament.

Entenem que els desenvolupadors necessiten:

Flexibilitat: Els PAM han demostrat que són flexibles a l'hora de proveir amb diversos mitjans d'autenticació, per tant si aconseguim portar l'essència dels PAM a l'entorn Windows, haurem assolit aquest objectiu.

Implantació: Tot i que la nostra plataforma podria ser distribuïda amb el mateix producte del desenvolupador de les aplicacions; seria idoni pel futur de la Plataforma que es pogués convertir en una aplicació estàndard en els entorns corporatius. Això només ho podem aconseguir fent que l'aplicació sigui útil i que proporcioni prou avantatges sobre solucions actuals per assolir una massa crítica d'instal·lacions.

Interfície Senzilla: Si la interfície per validar usuaris és fàcil de fer servir, augmentaríem la probabilitat de que en moltes noves versions d'aplicacions client/servidor es dediquin uns quants recursos a un nou sistema d'autenticació d'usuaris. Així aconseguiríem tenir una distribució d'aplicacions amb la capacitat latent abans de tenir una massa crítica instal·lada. Entenem per senzilla una interfície de programació amb pocs mètodes a cridar, mètodes clars i fàcilment comprensibles pel nom, amb funcions inequívoques i només amb els paràmetres necessaris.

Interfície Flexible: Tot i que la interfície ha de ser flexible, no podem prioritzar la flexibilitat a la senzillesa. Per tant hem d'intentar permetre el màxim de flexibilitat mantenint un nombre de paràmetres i crides coherents amb l'objectiu de la simplicitat.

Desenvolupadors de Mòduls

Els desenvolupadors de mòduls necessiten d'un entorn clar i unes interfícies ben documentades. A més l'objectiu de la Plataforma és permetre que desenvolupadors no professionals puguin desenvolupar nous mòduls per a entorns únics, per tant, potser s'ho repensarien si veiessin una interfície massa complexa.

El desenvolupador de mòduls necessita concentrar els esforços en implementar una solució segura, entre el proveïdor d'autenticació i l'aplicació que en farà ús. Per tant, complicar les *Module Programming Interface* (MPI) seria contraproduent per l'evolució de l'augment de la base instal·lada de la plataforma. Si proveïm d'una MPI simple reduïm les possibles fallades de comunicació entre la Plataforma i el proveïdor d'autenticació. D'aquesta manera si reduïm la probabilitat d'errors, contribuïm a una millor imatge pública de la plataforma, amb la qual cosa hi guanyen tots els actors: nosaltres com a creadors, el creadors de mòduls i aplicacions com a venedors de solucions i els administradors com a usuaris.

En aquest cas, al ser el públic objectiu també desenvolupadors, intentarem seguir el mateixos objectius que ens acabem de marcar en el cas dels Desenvolupadors d'Aplicacions.

Administradors de Sistemes

Els Administradors de Sistemes necessiten una solució, però aquesta ha de ser suportada pels fabricants de les eines que utilitzen. Doncs, el principal requeriment està fora del nostre abast, però podem ajudar al seu compliment assolint els requeriments detectats en els actors que hauran de treballar amb la Plataforma.

Els requeriments tècnics per als administradors són:

- **Flexibilitat:** La Plataforma ha de ser flexible amb les diverses aplicacions que en facin ús, i amb els proveïdors d'autenticació, de tal manera que no limiti el nombre d'opcions disponibles. Si el que vol un administrador és flexibilitat, l'idoni per ell és que cada aplicació pugui treballar amb tots els proveïdors d'autenticació existents. Hi hauria d'haver NxM opcions per arribar al producte cartesià entre aplicacions i proveïdors autenticació.
- **Configurable:** La plataforma ha ser fàcilment configurable tant amb eines desateses (*scripting*) com per eines d'usuari (GUI). A més s'han de proveir mètodes per tal que si hi ha més d'una aplicació en el sistema que usa la Plataforma, la configuració depenent de l'aplicació no vingui marcada pel fet que ja n'existeixi una altra configurada.
  Hem de proporcionar mètodes per mantenir la configuració de la Plataforma per a una aplicació, independent de la resta de configuracions de la Plataforma per a d'altres aplicacions. Així permetem que una sola instal·lació de la Plataforma pugui donar suport a totes les aplicacions configurades en el sistema.
- **Estabilitat:** La Plataforma amb tots els seus components ha de ser prou estable per poder suportar aplicacions critiques en la línia de negoci. En definitiva, s'ha de poder assegurar que si ha caigut una aplicació no ha estat pel fet d'haver introduït un nou actor en l'escena.
- **Lleugeresa:** La Plataforma ha de ser prou lleugera utilitzant els recursos, memòria i processador, perquè no sigui un factor a tenir en compte en el dimensionament de la màquina on s'executaran les aplicacions.

ESPECIFICACIÓ

Aquí especifiquem com interactuaran les diverses parts del sistema i quins drets i deures tenen cadascuna.

Casos d'ús

Abans de començar a especificar i decidir com s'ha de realitzar la Plataforma, hem de veure quins seran el Actors que la faran servir, i quins són els usos que li donaran.

Com que al llarg de la memòria seguirem el llenguatge gràfic de modelització *Unified Modeling Language* (UML), començarem aquí a introduir els primers diagrames UML, els Casos d'Ús.

Quan parlem de "Cas d'ús", ens referim a les circumstancies amb les que l'aplicació s'utilitzarà en gran part.

Un "Cas d'ús", com el seu nom indica, consisteix en modelar a molt alt nivell per a que s'utilitzarà un desenvolupament de *software* i qui el farà servir. Un diagrama d'aquest estil segons UML consisteix de tres components:

· **Actors**: són els diferents rols d'usuari amb que es pot trobar l'aplicació

· **Casos d'ús**: diferents funcions que cada Actor pot arribar a necessitar.

· **Comunicacions**: es relaciona cada "Cas d'ús" amb "l'Actor" que el fa servir.

Per la Plataforma hem considerat dos Actors, l'**Usuari** que accedeix a aplicacions que la utilitzen i l'**Administrador** que la configura, i activa o desactiva les aplicacions servidor que la fan servir.

Els casos d'ús considerats com a rellevants pel projecte són els següents:

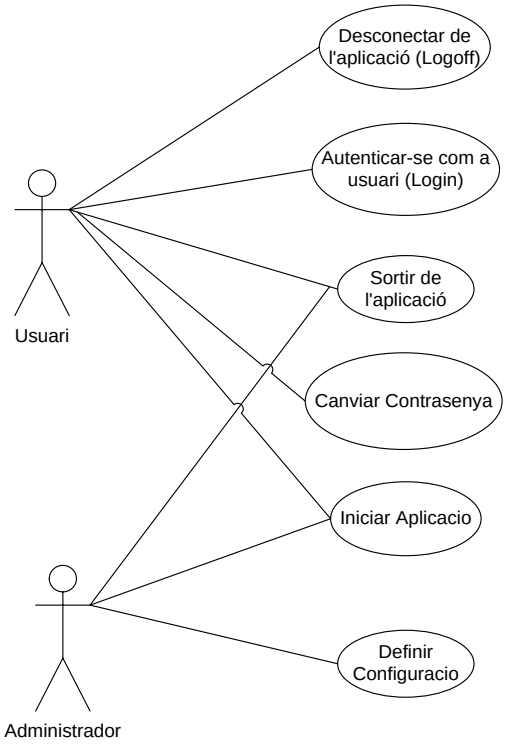


Figura 6

1. CAS D'ÚS: INICIAR APLICACIÓ

Actors: Usuari o Administrador (depèn dels tipus d'aplicació) (iniciadors)

Resum: L'actor inicia l'aplicació que utilitza la Plataforma, tant si és una aplicació servidora com si és una aplicació client o d'usuari.

Seqüència d'esdeveniments:

Accions dels Actors	*Resposta del Sistema*
1) L'actor executa les comandes necessàries per activar l'aplicació.	
	2) S'activa l'aplicació.
	3) S'inicialitza la Plataforma amb els valors configurats per a la instància de l'aplicació iniciada.

Opcions:

Error nº 1: L'aplicació no s'activa correctament per problemes aliens a la Plataforma. L'aplicació s'encarrega de notificar-ho.

Error nº 2: L'aplicació no s'activa per problemes amb la Plataforma. Aquesta notifica a l'aplicació de que s'ha produït un error per tal que aquesta ho notifiqui a qui correspongui.

2. CAS D'ÚS: AUTENTICAR-SE COM A USUARI (*LOGON*)

Actors: Usuari (iniciador)

Resum: Un cop iniciada l'aplicació, un usuari desitja utilitzar recursos proveïts per aquesta i s'ha d'autenticar per tenir-hi accés. L'aplicació demanarà a l'usuari que s'autentiqui i aquesta ho haurà de comunicar a la Plataforma.

Seqüència d'esdeveniments:

Accions dels Actors	Resposta del Sistema
1) L'actor proveeix de les dades necessàries (nom d'usuari/contrasenya) a l'aplicació.	
	2) L'aplicació comunica a la Plataforma les dades.
	3) La Plataforma intenta autenticar l'usuari utilitzant les dades passades per l'aplicació en el proveïdor d'autenticació seleccionat.
	4) La Plataforma comunica a l'aplicació que l'usuari ha estat autenticat i aquesta actua en conseqüència.

Opcions:

Error nº 1: L'usuari no ha estat validat correctament. L'aplicació és notificada del fet.

Error nº 2: No es pot establir comunicació amb el proveïdor d'autenticació. L'aplicació és notificada del fet.

3. CAS D'ÚS: CANVIAR CONTRASENYA

Actors: Usuari (iniciador)

Resum: En aplicacions que ho permetin, si l'usuari desitja canviar la seva contrasenya, ho podrà fer.

Seqüència d'esdeveniments:

Accions dels Actors	*Resposta del Sistema*
1) L'actor proveeix de les dades necessàries (nom d'usuari, contrasenya antiga i contrasenya nova) a l'aplicació.	
	2) L'aplicació comunica a la Plataforma les dades.
	3) La Plataforma intenta canviar la contrasenya a l'usuari en el proveïdor escollit
	4) La Plataforma comunica a l'aplicació que la contrasenya ha estat canviada amb èxit.

Opcions:

Error nº 1: La contrasenya antiga no era correcta. Es notifica a l'aplicació.

Error nº 2: No es pot establir comunicació amb el proveïdor d'autenticació. L'aplicació és notificada del fet.

Error nº 3: La nova contrasenya no compleix els requeriments del proveïdor d'autenticació. Es notifica a l'aplicació.

4. CAS D'ÚS: DESCONNECTAR-SE DE L'APLICACIÓ (*LOGOFF*)

Actors: Usuari (iniciador)

Resum: L'usuari decideix que ja no necessita fer ús dels recursos proveïts per l'aplicació. L'aplicació segueix activa. Per exemple, un servidor SGBD quan es desconnecta un client.

Seqüència d'esdeveniments:

Accions dels Actors	*Resposta del Sistema*
1) L'actor selecciona l'opció de desconnectar-se de l'aplicació.	
	2) L'aplicació comunica a la Plataforma la intenció de desconnectar la sessió d'usuari.
	3) La Plataforma notifica al proveïdor d'autenticació la desconnexió.
	4) La Plataforma comunica a l'aplicació que la desconnexió s'ha efectuat amb èxit.

Opcions:

Error nº 1: L'usuari no estava connectat. No es pot desconnectar.

Error nº 2: No es pot establir comunicació amb el proveïdor d'autenticació. L'aplicació és notificada del fet.

5. CAS D'ÚS: SORTIR DE L'APLICACIÓ

Actors: Usuari o Administrador (depèn dels tipus d'aplicació, client o servidor) (iniciadors)

Resum: L'Actor decideix que ja no són necessaris els serveis oferts per l'aplicació. En aquest cas d'ús la instancia de l'aplicació és tancada. Per exemple, al parar el servei d'un servidor SGBD.

Seqüència d'esdeveniments:

Accions dels Actors	*Resposta del Sistema*
1) L'actor selecciona l'opció de parar l'aplicació.	
	2) Si hi ha usuaris connectats, primer es desconnecten tots.
	3) L'aplicació comunica a la Plataforma la intenció de finalitzar l'aplicació
	4) La Plataforma tanca la comunicació amb el proveïdor d'autenticació i allibera els recursos utilitzats.

Opcions:

Error nº 1: La Plataforma no estava inicialitzada. No es pot finalitzar.

Error nº 2: No es pot establir comunicació amb el proveïdor d'autenticació. L'aplicació és notificada del fet.

6. CAS D'ÚS: DEFINIR CONFIGURACIÓ

Actors: Administrador

Resum: L'Actor actua sobre el sistema on està instal·lada la Plataforma per tal de parametritzar-la a la necessitats del seu negoci.

Seqüència d'esdeveniments:

Accions dels Actors	*Resposta del Sistema*
1) L'actor defineix els paràmetres en una interfície	
	2) Valida les dades entrades.
	3) Aplica els paràmetres en el dipòsit de configuració.
	4) Notifica a l'usuari de l'èxit en la operació.

Opcions:

Error nº 1: Les dades no són vàlides, notifica a l'actor.

Error nº 2: Problema al guardar la configuració, notifica del fet a l'actor.

El cas d'ús de *Definir Configuració* no el tractarem, ja que es podrà configurar amb les eines pròpies del sistema operatiu. Per tant no inclourem un Diagrama de Seqüència per a ell.

Diagrames de Seqüència

En aquesta secció mostrarem:

- els diagrames de seqüència dels diferents casos d'ús mostrats abans.
- la forma d'interaccionar tots els components entre si.

La comunicació Mòdul<->Proveïdor d'Autenticació està representada com un sol missatge, però depèn del protocol que implementi cadascun.

Un "Diagrama de seqüència" és un gràfic que ens mostra quina comunicació tenen els diversos mòduls que formen part del sistema en un determinat cas d'ús

Inici Aplicació

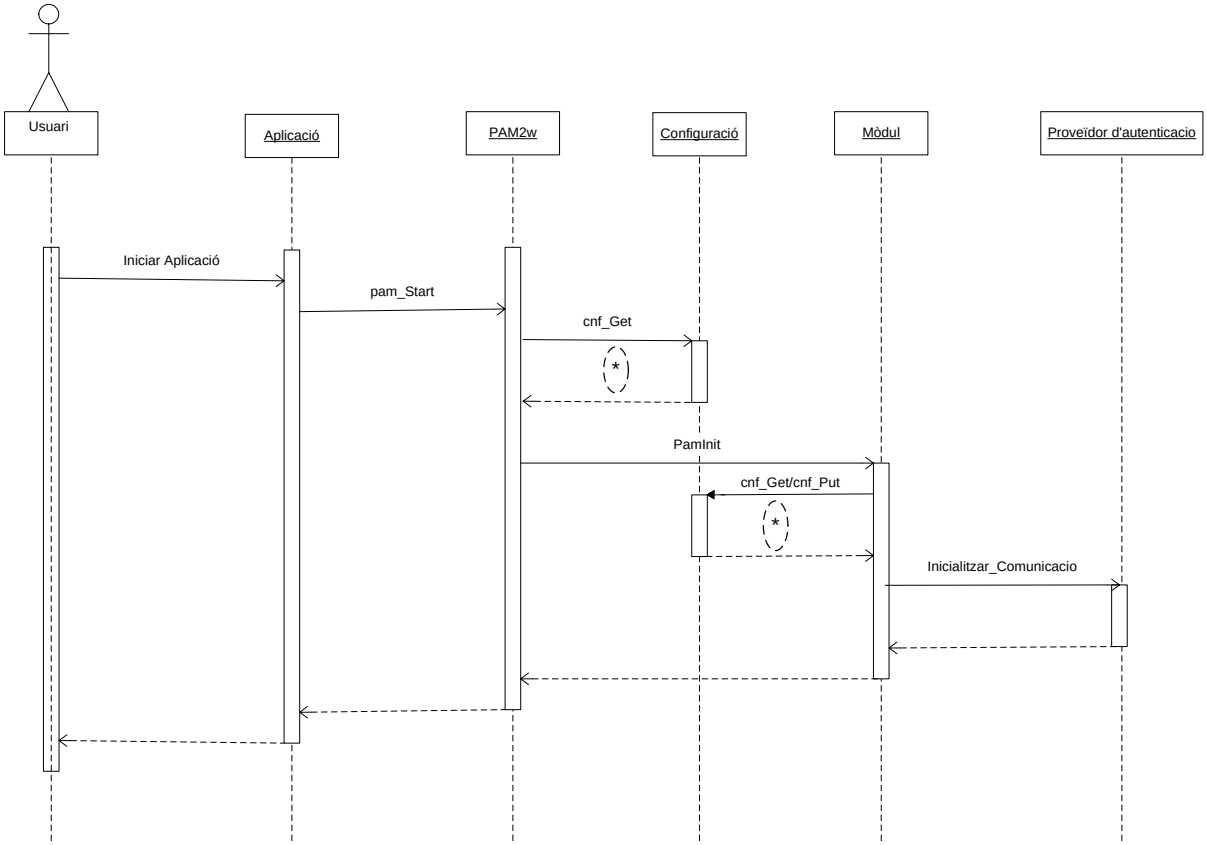


Figura 7

Al iniciar l'aplicació, aquesta inicialitza la plataforma que carregarà la configuració. Segons la configuració llegida, es carrega un o altre mòdul, que al seu torn revisarà la seva configuració segons s'escaigui o no. Depenent del protocol implementat pel mòdul es prepararà la comunicació vers el proveïdor, o com a mínim comprovar que tot estigui correcte.

Autenticar-se com a Usuari (Logon)

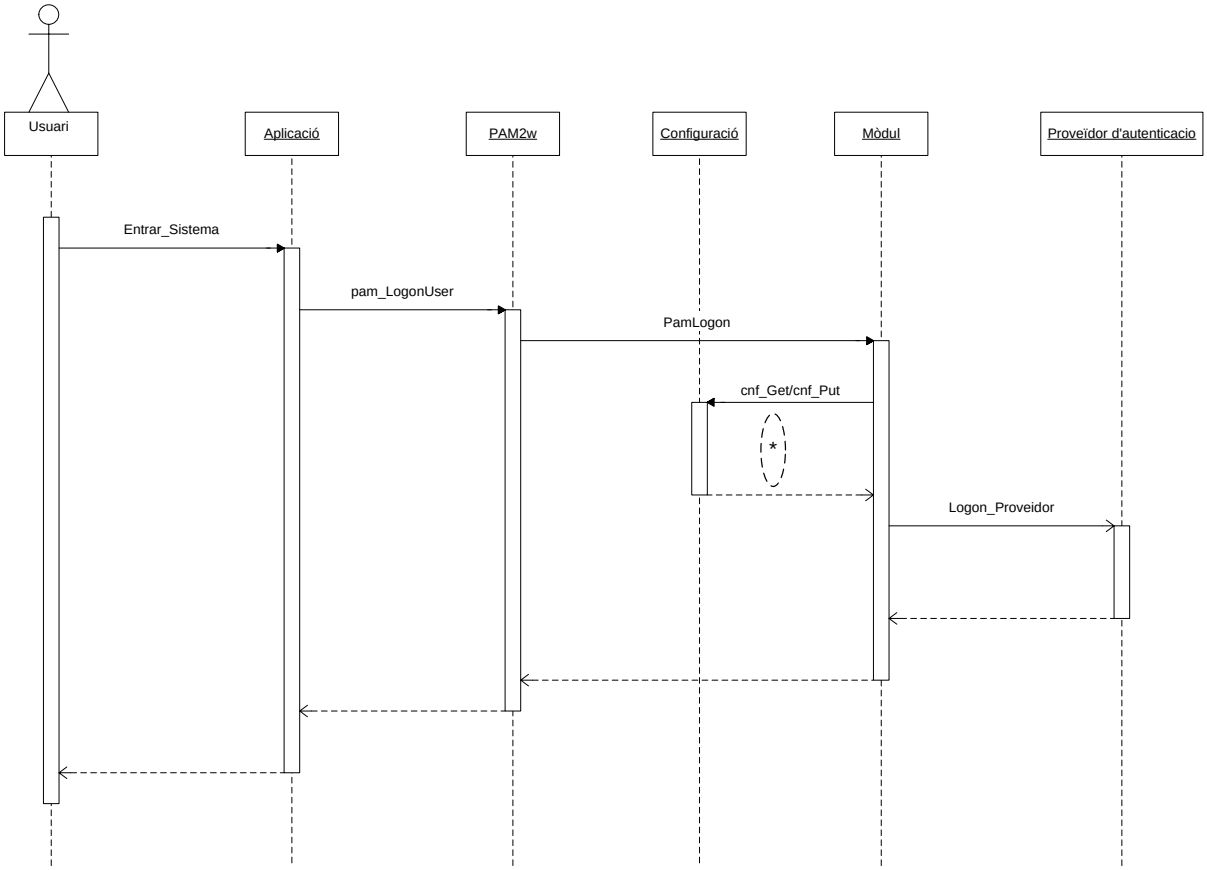


Figura 8

Quan un usuari intenta accedir a una aplicació que faci ús de la Plataforma, l'aplicació envia un missatge a la Plataforma indicant que aquell usuari vol entrar. Aquesta passa la requesta al mòdul pertinent, que comprova si tot es correcte. Tot seguit, el mòdul intenta validar la petició d'accés amb el proveïdor d'autenticació. Un cop autenticat l'usuari, es delega l'autorització del mateix a l'aplicació, és a dir, sabem que l'usuari és qui diu que és, però no sabem si l'hem de deixar entrar o no segons les restriccions imposades per l'aplicació. Per exemple: horari d'accés, lloc d'accés, etc.

Canvi Contrasenya

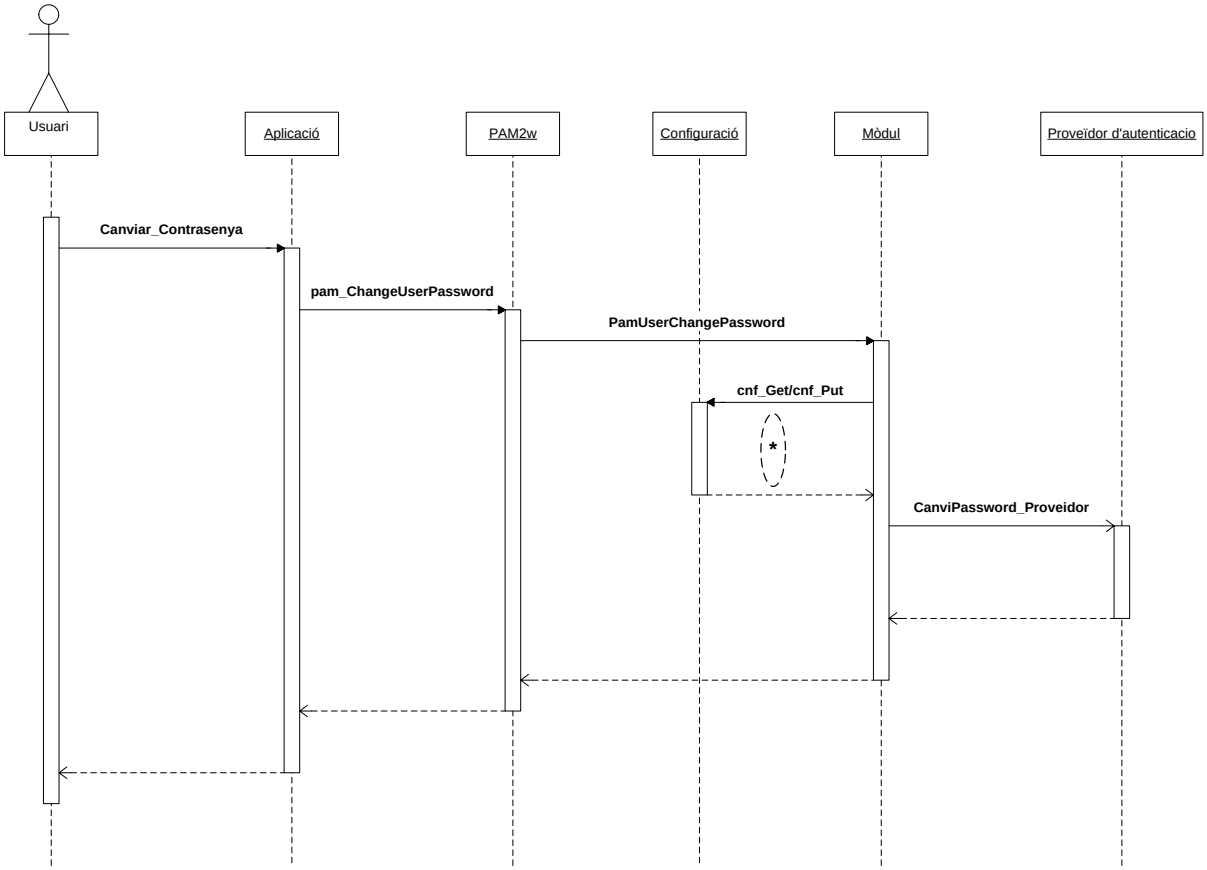


Figura 9

Si l'aplicació disposa de mitjans pel canvi de contrasenya, demanarà la contrasenya anterior i la nova, passarà la requesta del canvi a la Plataforma. Aquesta última la passarà al mòdul, que comprovarà la validesa o altres paràmetres dependents d'ell i intentarà canviar la contrasenya en el proveïdor d'autenticació associat.

Desconnectar-se de l'Aplicació

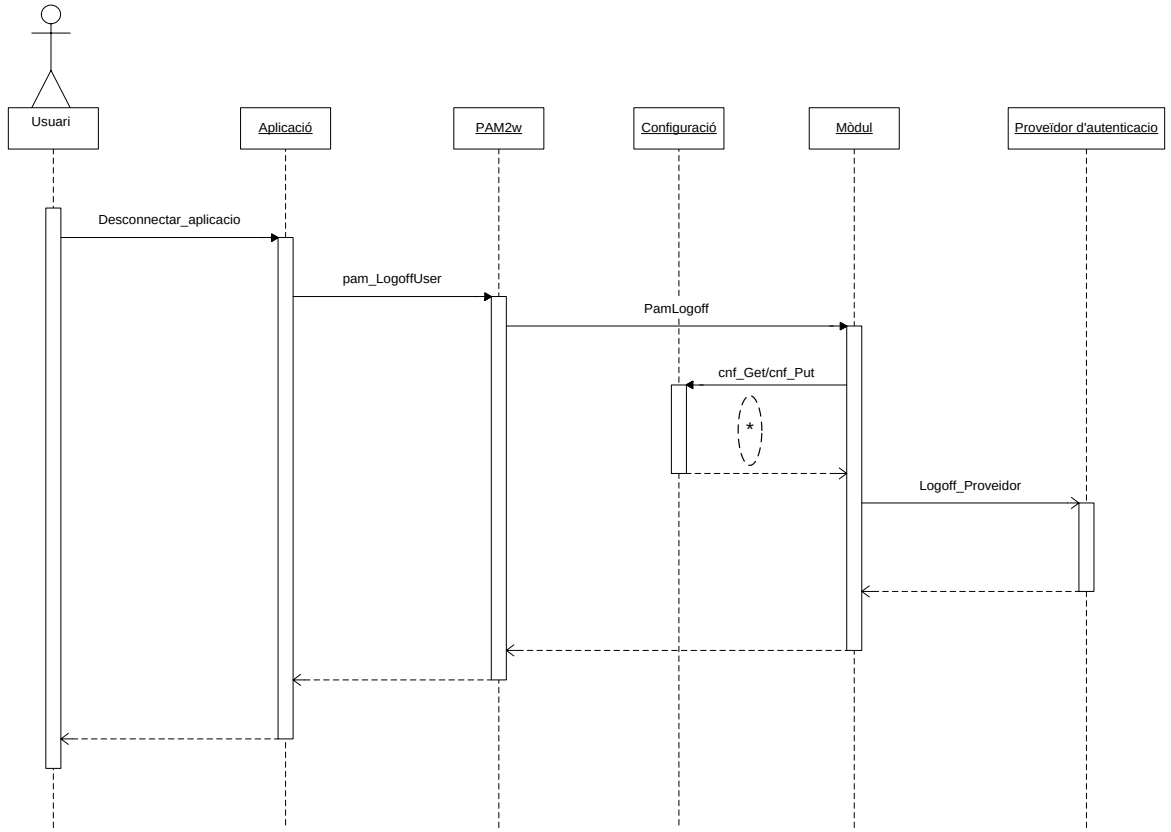


Figura 10

Quan l'usuari decideix tancar la sessió, depenent del proveïdor d'autenticació, pot ser recomanable notificar el fet. Per tant l'aplicació ha d'avisar a la Plataforma, que al seu torn ho farà al mòdul i aquest depenent de la configuració ho notificarà o no al proveïdor. A més durant aquest procés s'hauria d'alliberar tota la informació relativa a la sessió.

Sortir de l'Aplicació

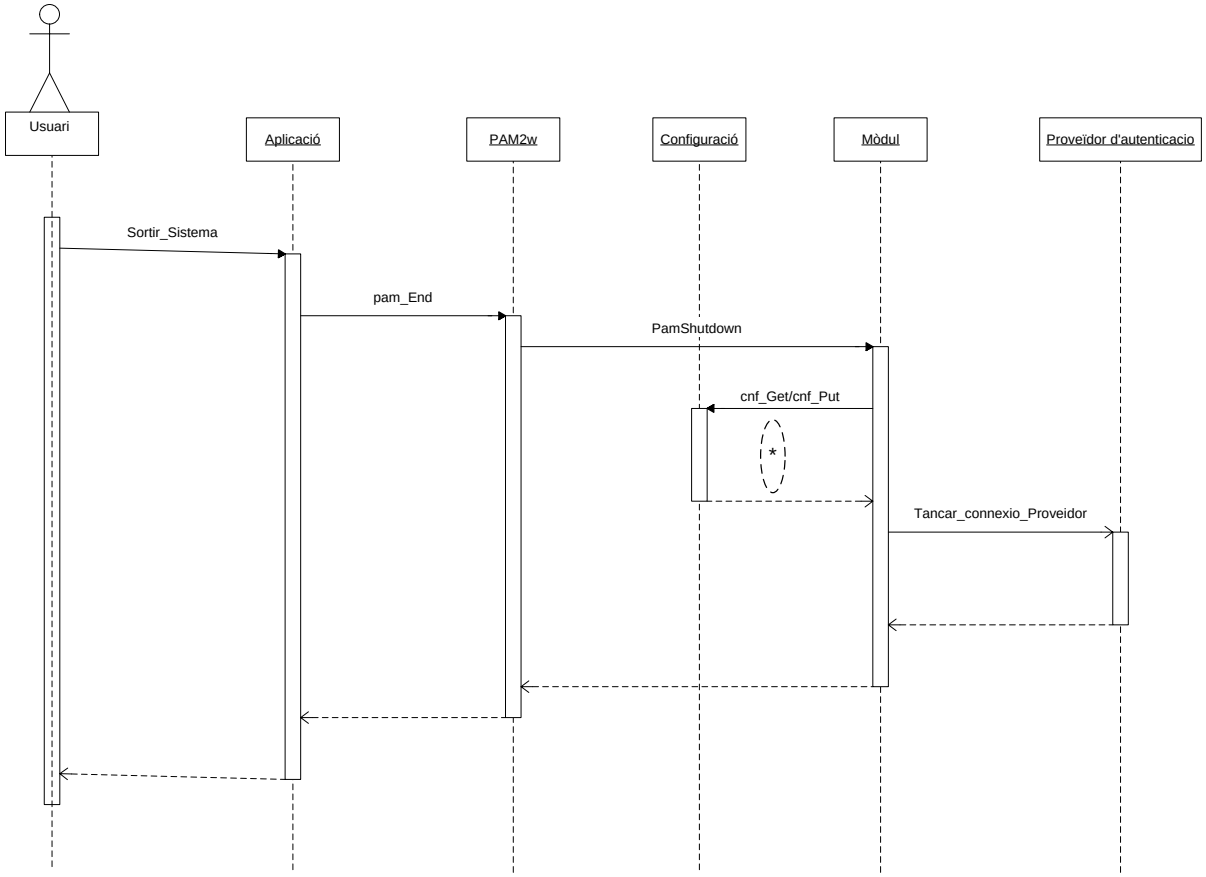


Figura 11

Quan l'usuari decideix tancar l'aplicació, s'ha d'alliberar tota la memòria i tancar ordenadament totes les sessions obertes en aquell moment.

DISSENY

Com serà l'arquitectura interna de cada una de les parts per tal d'acomplir els requisits del capítol anterior.

Arquitectura

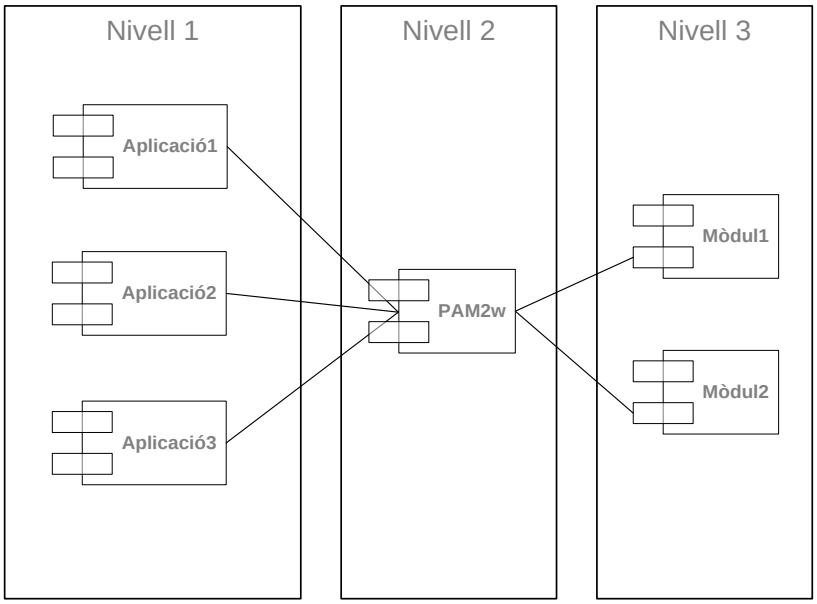
La Plataforma d'Autenticació Modular per a Microsoft Windows (PAM^{2}w) està estructurada en tres nivells.

- El **primer nivell** és el que interacciona amb l'usuari. A partir d'ara anomenarem a aquest nivell, "Aplicació". Aquí es demana a l'usuari que introdueixi les dades per tal de realitzar l'autenticació, se li mostren missatges d'estat o d'error i s'implementa tota la lògica de l'aplicació que utilitza la Plataforma. En definitiva, és tota la interacció amb l'usuari. Per exemple: *GInA*, servidor FTP, servidor SSH, aplicació de Recursos Humans, ERP, etc.
- El **segon nivell** és el nexe d'unió entre els altres nivells i és pròpiament la Plataforma. Consisteix en una Dynamic–Link Library (DLL) [MS01] que es carrega a l'espai d'adreces de l'Aplicació, llegeix la configuració comuna de la plataforma i l'específica per la instància de l'aplicació. D'acord amb això carrega els mòduls necessaris i els inicialitza. En aquest nivell, es proporcionen els mitjans per realitzar entrades en el registre de successos de Windows i passar informació de l'Aplicació al Mòdul i a l'inrevés.
- En el **tercer nivell** hi ha els mòduls amb els que s'implementa la comunicació (protocol) cap al proveïdor d'autenticació, ja sigui un fitxer de text amb les contrasenyes, o la comunicació cap a un servidor que faci de proveïdor d'autenticació (Ldap, Radius, etc.)

Què és una DLL?

Una *Dynamic Link Library* (DLL) és un fitxer amb codi compilat on s'emmagatzemen implementacions de funcions que es comparteixen. Quan una aplicació necessita d'alguna d'aquesta funció carrega aquest fitxer en el seu espai de memòria de tal manera que pot accedir a les funcions.

L'objectiu de les DLLs és aprofitar les bondats de la compartició de codi (un canvi s'aprofita en totes les aplicacions que la fan servir) o bé per tenir diverses implementacions de les mateixes funcions i així poder utilitzar codi de tercers (*plugins*).



Drets i deures de cada Nivell

Tot seguit anem a descriure els contractes que té cada nivell, és a dir, quines són les responsabilitats que té cadascun vers els altres.

Nivell 1

- inicialitzar correctament la plataforma i proporcionar els paràmetres per poder utilitzar el sistema d'autenticació que ha configurat l'administrador del sistema.
- mostrar o enregistrar els missatges que generin tant la plataforma com els mòduls. Realitzant les tasques d'interfície vers l'usuari de la plataforma, ja sigui en mode gràfic, text o emmagatzemant a fitxer.
- comprovar que totes a les variables a que accedeix són correctes. La PAM^{2}w no garanteix que tots els punters estiguin inicialitzats.

Nivell 2

- carregar el/s mòdul/s necessaris segons la configuració de l'aplicació hoste.
- proporcionar serveis d'accés a la configuració de la Plataforma.
- enregistrar els missatges dels diversos components al registre de successos.

Nivell 3

- accedir correctament al sistema d'autenticació (fitxer, Ldap, etc.).
- intentar omplir tota la informació de l'usuari (nom, cognom, directori personal, grup a que pertany, etc.) a partir de la implementació per part del proveïdor d'autenticació o gràcies a la configuració estàtica proporcionada per l'administrador del sistema.

Components del Sistema

Després d'enumerar les relacions de cada Nivell vers els altres, toca exposar com es plasmaran aquests Nivells en el sistema on s'executaran.

Nivell 1

El Nivell 1, el de l'Aplicació contindrà tots els components necessaris per tal que l'aplicació clienta de la Plataforma disposi de totes les funcionalitats vers l'usuari que s'esperen. Per exemple en el cas d'un programa de comptabilitat, tots els components, fitxers i d'altres necessaris perquè independentment de quin usuari hagi accedit totes les funcionalitats (assentaments, formularis, càlculs, etc.) s'executin correctament.

Nivell 2

El Nivell de la Plataforma necessitarà dels components i fitxers necessaris per tal que el Nivell 2 pugui ser el nexe d'unió entre l'Aplicació i els Mòduls. Per tant necessitem un fitxer de càrrega dinàmica que contingui la tota la lògica per tal d'interaccionar amb l'Aplicació, carregar el mòdul apropiat i emmagatzemar els successos originats tant pels mòduls com per la pròpia Plataforma.

Per aquest nivell tindrem un fitxer anomenat ***pam2w.dll*** on hi haurà totes les funcions necessàries per realitzar les tasques encomanades al nivell. En el Nivell 2 també hi tindrem els fitxers on es guardi la configuració (el registre en aquesta versió) i on desem els esdeveniments que volem guardar (a esdeveniments de Windows en aquesta versió).

Nivell 3

El Nivell dels Mòduls serà l'encarregat d'Autenticar l'usuari en el proveïdor d'Autenticació elegit, per tant haurem de tenir disponibles tots els fitxers necessaris per tal que la comunicació cap al proveïdor pugui ser establerta amb èxit. El nombre necessari i tipus de fitxers dependrà en major part de com s'ha implementat el Mòdul que de quin sistema d'autenticació s'empra.

En principi tindrem fitxers anomenats ***Modul<prov-Autenticació>.dll***, per exemple en el cas de proveïdor Ldap, ***ModulLdap.dll***. Tot i que la Pam2w proporciona mitjans per guardar la configuració de forma centralitzada, els mòduls poden utilitzar els mitjans convenients per guardar la configuració (registre, fitxer de text, xml [RFC3076], etc) tot i que per facilitat de gestió es recomana fer servir només les eines de que proveeix la PAM2w.

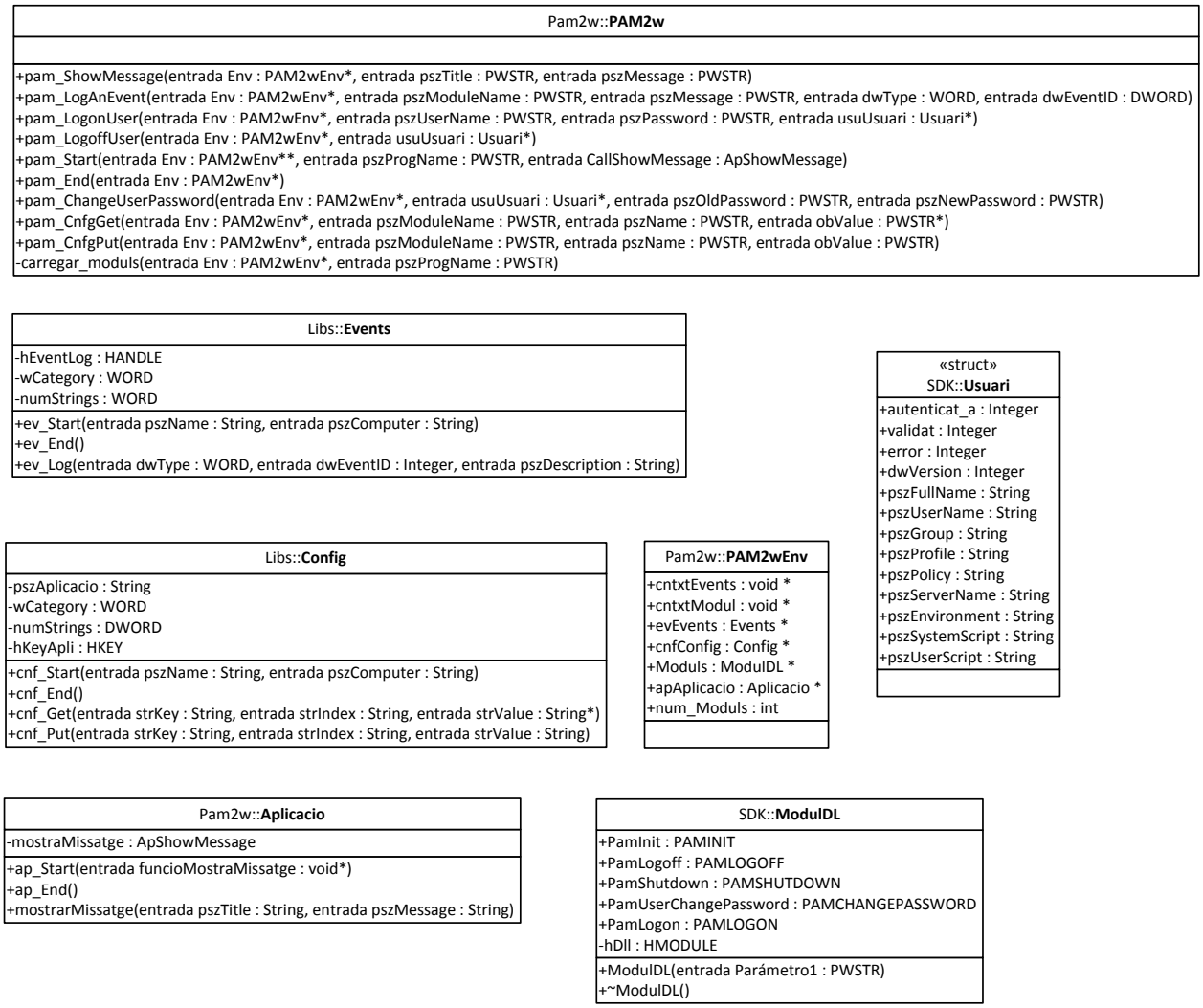
Diagrama de Classes

Pel diagrama de classes hem de separar els diversos nivells de la Plataforma, les eines de desenvolupament i els exemples que aportem amb la distribució estàndard.

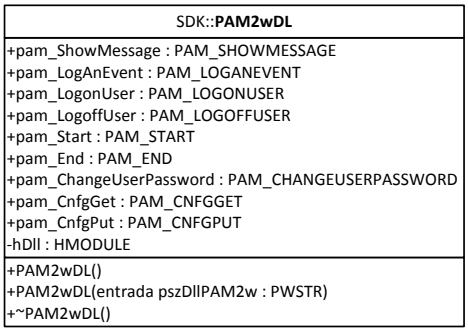
Per una informació més amplia veure l'Apèndix A.

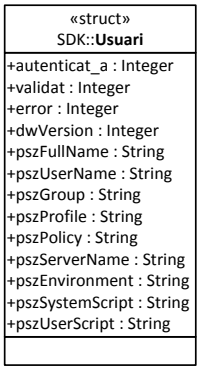
Primer descriurem el segon nivell, el nucli de la plataforma i com es relacionen les classes entre si.

Aquí tenim una classe principal, la *Pam2w*, dues que defineixen tipus de dades i no disposen de mètodes d'accés, la *Pam2wEnv* i la *Usuari*, dues auxiliars que implementen el metode d'acces a configuració i el registre d'esdeveniments, la *Config* i *Events*, i dues més que permeten al segon nivell iniciar comunicacions cap al primer, *Aplicacio*, i cap al tercer, *ModulDL*.



Un dels casos més simples de comentar és el de les aplicacions, ja que només han d'incloure una Classe en el seu codi i cridar als mètodes especificats i un tipus de dades per tal de poder rebre les dades inicialitzades per la Plataforma. La classe en qüestió és la *Pam2wDL* (Pam2w Dynamic Loader) i el tipus de dades l'*Usuari.*

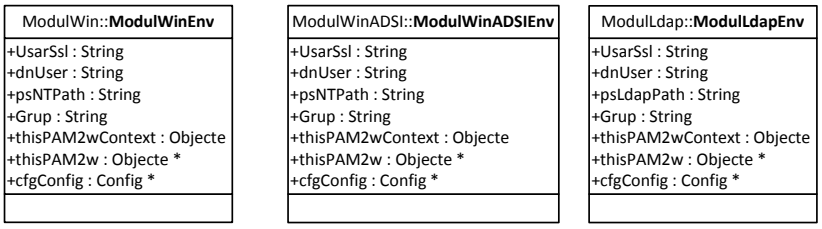


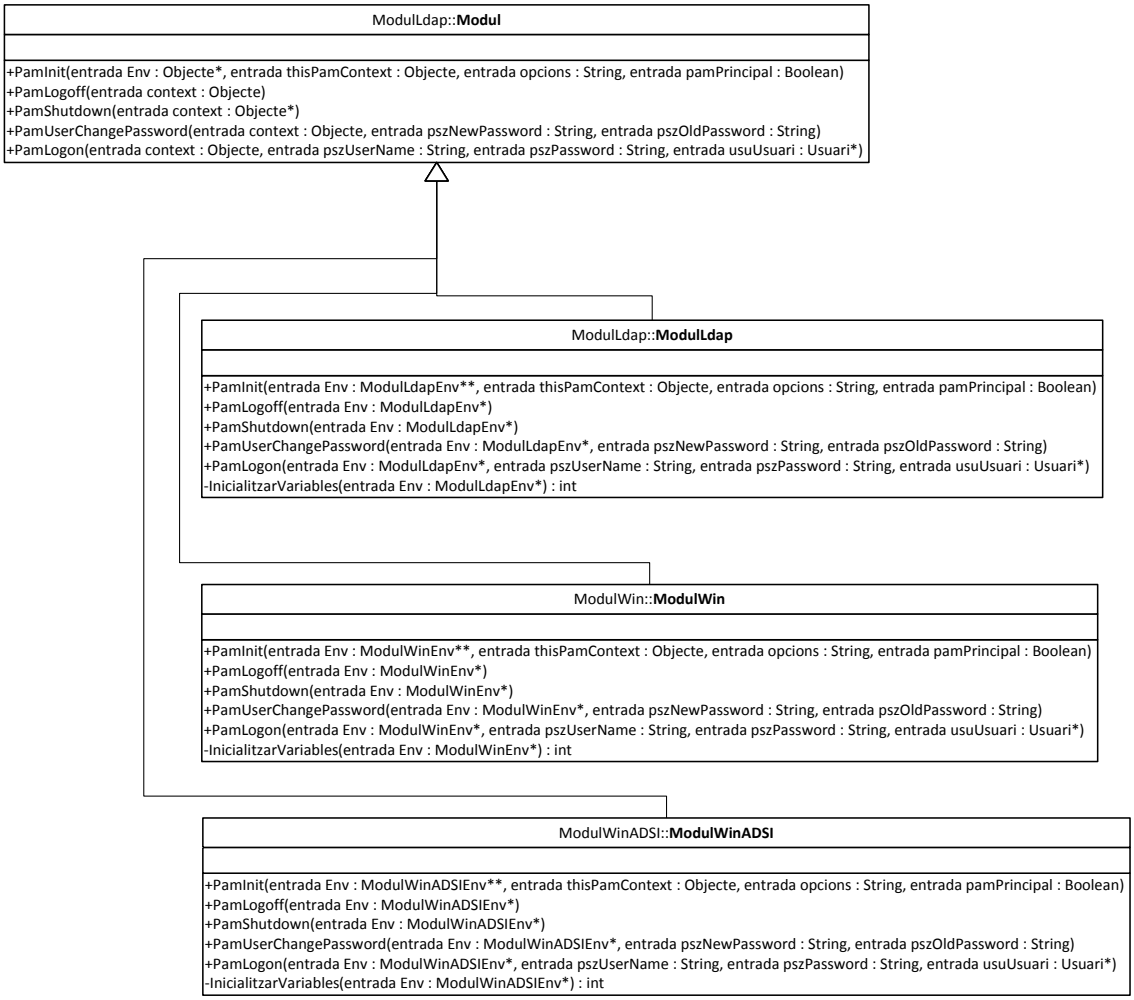


Per últim però no menys important, ens queda el tercer nivell, els mòduls. Hem decidit que la millor forma de mantenir una interfície consistent entre els mòduls que implementen diversos proveïdors d'autenticació era aprofitar el concepte d'herència i recomanar a tots els desenvolupadors a que el seu mòdul hereti de la classe *Modul* i s'abstinguin de redefinir mètodes. Els mòduls també necessiten d'una estructura de dades, però tot i que hi ha recomanacions ja és decisió del desenvolupador quina estratègia fer servir.

La informació d'estat necessària per a cada nivell s'ha d'especificar en cada crida per tal de poder tolerar múltiples mòduls, crides o usuaris dins de la mateixa aplicació. Per tant el primer paràmetre de cada crida sempre és una referència a un punter que el nivell adequat haurà inicialitzat correctament i serà el responsable d'alliberar la memòria assignada quan sigui el moment. Aquesta gestió de la memòria fer forma compartida, és una de les diferencies més importants entre una implementació d'una aplicació monolítica i una altra modular. L'altra diferència és que necessitem unes classes patró on el seu mètode constructor sigui el carregar la biblioteca de mètodes a memòria i inicialitzar els atributs oportuns.

Tot seguit, el diagrama de classes dels tres exemples que hem desenvolupat per fer les proves.




ModulLdap::**Modul**
+PamInit(entrada Env : Objecte*, entrada thisPamContext : Objecte, entrada opcions : String, entrada pamPrincipal : Boolean)
+PamLogoff(entrada context : Objecte)
+PamShutdown(entrada context : Objecte*)
+PamUserChangePassword(entrada context : Objecte, entrada pszNewPassword : String, entrada pszOldPassword : String)
+PamLogon(entrada context : Objecte, entrada pszUserName : String, entrada pszPassword : String, entrada usuUsuari : Usuari*)
ModulLdap::**ModulLdap**
+PamInit(entrada Env : ModulLdapEnv**, entrada thisPamContext : Objecte, entrada opcions : String, entrada pamPrincipal : Boolean)
+PamLogoff(entrada Env : ModulLdapEnv*)
+PamShutdown(entrada Env : ModulLdapEnv*)
+PamUserChangePassword(entrada Env : ModulLdapEnv*, entrada pszNewPassword : String, entrada pszOldPassword : String)
+PamLogon(entrada Env : ModulLdapEnv*, entrada pszUserName : String, entrada pszPassword : String, entrada usuUsuari : Usuari*)
-InicialitzarVariables(entrada Env : ModulLdapEnv*) : int
ModulWin::**ModulWin**
+PamInit(entrada Env : ModulWinEnv**, entrada thisPamContext : Objecte, entrada opcions : String, entrada pamPrincipal : Boolean)
+PamLogoff(entrada Env : ModulWinEnv*)
+PamShutdown(entrada Env : ModulWinEnv*)
+PamUserChangePassword(entrada Env : ModulWinEnv*, entrada pszNewPassword : String, entrada pszOldPassword : String)
+PamLogon(entrada Env : ModulWinEnv*, entrada pszUserName : String, entrada pszPassword : String, entrada usuUsuari : Usuari*)
-InicialitzarVariables(entrada Env : ModulWinEnv*) : int
ModulWinADSI::**ModulWinADSI**
+PamInit(entrada Env : ModulWinADSIEnv**, entrada thisPamContext : Objecte, entrada opcions : String, entrada pamPrincipal : Boolean)
+PamLogoff(entrada Env : ModulWinADSIEnv*)
+PamShutdown(entrada Env : ModulWinADSIEnv*)
+PamUserChangePassword(entrada Env : ModulWinADSIEnv*, entrada pszNewPassword : String, entrada pszOldPassword : String)
+PamLogon(entrada Env : ModulWinADSIEnv*, entrada pszUserName : String, entrada pszPassword : String, entrada usuUsuari : Usuari*)
-InicialitzarVariables(entrada Env : ModulWinADSIEnv*) : int

EINES I METODOLOGIA D'IMPLEMENTACIÓ

En aquest capítol discutirem quines eines, metodologies i tecnologies triem de les que tenim disponibles per a implementar la solució a que hem arribat al capítol anterior.

Implementació

En aquest capítol comentarem les diverses decisions de disseny que s'han pres per tal d'implementar la Plataforma.

Dinamic Load Libraries (DLLs)

S'ha escollit implementar la Plataforma com un conjunt de DLLs ja que ens proporciona una alta flexibilitat al moment d'execució. D'aquesta manera podem estendre l'arquitectura del disseny de la Plataforma al sistema del fitxers, conservant tota la flexibilitat que caracteritza a una arquitectura de tres Nivells.

Una altra raó d'escollir DLLs estàndard ha estat el fet de que són independents del llenguatge, es poden implementar en qualsevol dels entorns de desenvolupament disponibles per a Windows. Així, cada component de la Plataforma, aplicació, PAM^{2}w o mòdul, es poden implementar fent servir l'entorn més adequat per a la tasca o bé amb el que està més preparat el desenvolupador.

Espai de Memòria

Hem triat utilitzar l'espai de memòria de l'aplicació que fa servir els serveis de la Plataforma tant per seguretat com eficiència.

Seguretat: El fet que la Plataforma comparteixi memòria amb l'aplicació significa que la comunicació entre elles està protegida pel sistema operatiu i només es podria interceptar amb un *debugger*, per tant l'única forma d'interceptar les comunicacions passa per tenir els privilegis necessaris per adjuntar un *debugger* a l'aplicació. D'aquesta manera també evitem la necessitat d'establir mitjans d'autenticació mútua, d'encriptació del canal o de les comunicacions, ja que si es pot interceptar la comunicació entre la Plataforma i l'aplicació també es podrà accedir a les dades desencriptades tant si en té el control l'aplicació com si el té la Plataforma

Eficiència: Al ser una solució en memòria compartida ens estalviem tota la sobrecàrrega de pas de paràmetres i alhora la de comprovació i validació dels accessos a les interfícies exposades. Deixant de fer aquestes comprovacions no obrim noves vulnerabilitats al sistema, ja que la Plataforma només serà vulnerable a l'Aplicació o al Proveïdor d'Autenticació, hem de confiar en ambdós. Si el proveïdor d'autenticació no és fiable, no serà un proveïdor d'autenticació vàlid (els Mòduls podrien implementar autenticació mútua) i si l'aplicació fa mal ús de les dades que ens ha proporcionat estarà fent un ús no legítim de dades que ja disposava, per tant és molt més senzill fer-les servir abans de passar-les a la Plataforma que no pas intentar recuperar les dades de forma il·lícita.

Stubs

Per tal de fer possible la carrega dinàmica dels diversos components, s'han implementat una sèrie de *Stubs* (trossos) que contenen el codi per carregar *plugins*. Són les classes amb sufix *DL*.

Aquestes classes ofereixen els mateixos mètodes que ofereixen les classes "reals" però sense incloure el codi que realitza la funció, només tenen l'estrictament necessari per tal que el compilador no es queixi i puguin cridar al mètode real un cop estigui tot inicialitzat.

El constructor d'aquestes classes carrega en l'espai d'adreces de l'aplicació la DLL especificada i vincula els seus mètodes als mètodes proporcionats per la última.

Amb aquesta solució aconseguim que els desenvolupadors de mòduls siguin independents dels desenvolupadors de la Plataforma i a l'inrevés, només han de seguir l'API definit i és decisió de l'usuari fer servir un o altre Mòdul per una determinada Aplicació.

Components al Sistema de Fitxers

Els diversos Nivells de la Plataforma es veuen reflectits en una sèrie de fitxers al disc de la màquina.

Nivell 1

Aquest és el nivell Aplicació, depenent de la complexitat o des les tecnologies utilitzades estarà composta de més o menys fitxers, la majoria dels quals estaran localitzats a la carpeta "*Archivos de Programa*" de la màquina.

En les aplicacions d'exemple de la Plataforma tenim el Pcqcp.exe i el Pam2wGina.dll.

Nivell 2

Aquí hi tenim la Plataforma, en el mateix fitxer, *pam2w.dll* s'ha implementat tota la lògica del nus de comunicacions entre Aplicació i Mòdul, els serveis de lectura de configuració i de registre d'esdeveniments.

En aquest nivell es podrien incloure la resta de fitxers que donen suport a la plataforma i que s'instal·len en tots els sistemes Windows per defecte.

Nivell 3

El tercer nivell, és el nivell dels Mòduls. Cada mòdul implementa la comunicació amb un proveïdor d'autenticació diferent. Així tenim que cada un és un fitxer en format DLL 32 bits de Windows amb el seu nom i els fitxers d'ajut que pot necessitar.

Els Mòduls són la part de la Plataforma que amb més assiduïtat tindrà dependències amb fitxers externs, ja que la majoria de vegades no implementarem tota la lògica per interactuar amb els proveïdors d'autenticació al aprofitar codi realitzat per tercers que encapsulin el protocol a baix nivell.

Eines disponibles i Elecció

Llenguatges de Programació

Java

Aquest llenguatge de programació és propietat de Sun MicroSystems. Està disponible per a una gran quantitat d'entorns diversos i el principal avantatge que ofereix als desenvolupadors, és que garanteix que una aplicació funcionarà a qualsevol dels entorns pel quals està disponible. És un llenguatge orientat a objectes pur, fortament tipat. Al estar basat en màquina virtual, és força complicat l'accés al sistema operatiu i la comunicació amb d'altres aplicacions.

Visual Basic

És un entorn de desenvolupament visual amb un llenguatge basat en el Basic. És molt útil per fer prototipatges o aplicacions senzilles amb força carrega visual (GUIs). En canvi, és també bastant complicat poder accedir al sistema i interactuar amb d'altres aplicacions.

C++

És l'estàndard *de facto* en llenguatges de programació orientats a objectes tot i no ser en sí mateix un llenguatge orientat a objectes pur. Una de les principals característiques de C++ és que tot i ser bastant orientat a objectes permet obviar moltes de les restriccions que imposaria un llenguatge fortament tipat.

Elecció

Ens hem decantat per C++ ja que és el llenguatge que més facilitat ens donava a la integració amb d'altres aplicacions i amb el propi sistema. Tot i que té les seves mancances formals, les característiques operatives el fan fer molt superior, sobretot pel fet de no estar basat en una maquina virtual com els altres, permet amb molta facilitat carregar els mòduls a l'espai de memòria de l'aplicació.

Eina de Desenvolupament

MinGW: Minimalist GNU for Windows [MIN01]:

És un entorn de desenvolupament per a Windows basat en eines de desenvolupament lliures sota el paraigües de la GNU. Bàsicament és un conjunt d'eines que permeten generar fitxers executables sobre Windows sense necessitat de tenir fitxers de suport externs (*ala* Cygwin [CY01]). No es pot considerar un entorn de desenvolupament integrat ja que li manca una eina de lligui totes les altres, per tant tot el desenvolupament s'ha de realitzar amb eines desconnectes, perdent eficiència. Un altre problema és que és un entorn en desenvolupament i parts crítiques de la Plataforma com ara seria el suport de COM [MS02], no estan del tot acabades. Per tant és un bon entorn gratuït i lliure per a desenvolupar sota Windows, sempre i quant el desenvolupament no hagi de tractar amb serveis que pugui oferir la plataforma de Microsoft.

Borland C++ Builder [BO01]:

Era l'entorn de desenvolupament C++ estàndard fa uns quants anys, amb potser el millor compilador del mercat, però de mica en mica l'entorn de desenvolupament de Microsoft (Visual Studio) es va anar cruspint el mercat, bàsicament pel poc suport que oferia el C++ Builder a les noves tecnologies que Microsoft introduïa als seus productes. Aquest és un defecte que encara conserva el C++ Builder, ja que va una mica darrera de Microsoft en funcionalitats.

Microsoft Visual Studio [MS03]:

És l'entorn estàndard de desenvolupament sobre Windows. Han anat afegint funcionalitats per tal que en la generació de codi sigui un dels entorns més còmodes, té suport per totes les tecnologies introduïdes per Microsoft a l'entorn Windows. També ofereix un entorn de depuració de codi generat molt útil i la documentació sobre el sistema i els exemples són força clars. Pel que respecta a la qualitat del codi generat pel compilador ha anat millorant molt i actualment ja és un dels millors, optimitzant codi per les diverses arquitectures de processadors presents.

Elecció:

Ja que hem escollit desenvolupar la Plataforma en C++, haurem d'escollir un entorn de desenvolupament que ens permeti ser productius, no tingui problemes al generar codi partint de C++ i suporti gran part de les tecnologies que ens ofereix l'entorn Windows NT/2000. Per tant l'elecció és força clara, l'entorn **Microsoft Visual Studio** és el més adient i compleix tots els requeriments per poder realitzar aquesta tasca.

CONCLUSIÓ

Balanç del Projecte, a nivell econòmic, tècnic i personal.

Objectius Assolits

Hem aconseguit assolir objectius marcats en l'informe preliminar del Projecte.

Aquests eren:

- Analitzar el mercat.
- Trobar solució efectiva en despeses per l'autenticació en entorns heterogenis.
- Especificar una *Application Programming Interface* (API).
- Especificar una Module Programming Interface (MPI).
- Definir els *Includes* i llibreries necessàries per desenvolupar mòduls i aplicacions basats en la Plataforma.
- Desenvolupar la Plataforma amb carrega dinàmica.
- Trobar solució als problemes que han anat sorgint mentre realitzàvem el Projecte.
- En Software Development Kit (SDK)
    - Exemple d' aplicacions que fan servir la Plataforma.
        - PcqcPAM2w
        - Pam2wGina
    - Exemples de Mòduls.
        - ModulLdap
        - ModulWin
        - ModulAdsiWin

Recursos Usats

Recursos humans:

- un projectista.

Recursos tècnics:

- Llicencia de Microsoft Visual Studio.
- Accés a la MSDN Library com a documentació.
- Entorn de desenvolupament.
  - Màquina Windows XP amb eines de desenvolupament i depuració
- Entorn de test.
  - Màquina Windows XP amb Vmware Server
  - Màquina virtual Windows 2003 Server en funcions de controlador de domini
  - Màquina virtual Windows 2003 Server en funcions de servidor de terminal
  - Màquina virtual Linux Fedora com a servidor LDAP

Línies de Treball Obertes

Encara que hem intentat atacar el que hem entès més important, queden certes línies de treball obertes, les quals citem a continuació:

- Seguir desenvolupant la Plataforma de tal manera que arribi a proveir els quatre estils dels PAM (Autenticació, Comptes, Sessió, Contrasenya), i no tant sols autenticació i contrasenya com ara. D'altre banda, s'ha escollit aquestes dues vessants per ser les mes útils en els entorns en els quals s'implantarà.
- Desenvolupar una eina gràfica pels administradors que permeti configurar la Plataforma de manera centralitzada.
- Ampliar el perfil de l'usuari. Basant-nos en aquest Projecte, podem validar positiva o negativament a un Usuari segons la informació del proveïdor d'autenticació. Però es podria arribar a obtenir informació sobre polítiques de grup, directoris per defecte, servidor de correu associat al usuari, adreça IP a configurar a la màquina i un llarg etcètera. De totes maneres la part desenvolupada per aquest projecte és la que proveeix de la infraestructura necessària per facilitar l'extensió del perfil.
- Adaptar aplicacions existents (o futures) a la Plataforma. No s'ha fet, perquè adaptar totes les aplicacions existents, surt de l'objectiu del projecte, i segon és inabordable amb els recursos disponibles.
- Ampliar el catàleg de mòduls disponibles per la Plataforma. D'aquesta manera, s'oferiria un ampli ventall de protocols d'autenticació utilitzables amb la Plataforma.

  Mòduls disponibles inclosos amb el Projecte:

  - Ldap
  - Windows Nadiu
  - Windows Active Directory

  Mòduls que podrien ser introduïts en la Plataforma creada:

  - text pla tal com Unix
  - Radius
  - SQL
  - i un llarg etc.

Valoració Personal

Durant aquest projecte he aprés diverses coses, algunes d'aplicació a la meva vida acadèmica, d'altres a la professional i per últim també a la meva vida personal.

Ha estat un projecte llarg per diversos motius, un per la complexitat i la falta de coneixements de que disposava en el desenvolupament en el món Windows, per la manca de documentació i sobretot exemples de modificacions de processos interns de Windows, però el que ha provocat més dilació, que no càrrega de treball, ha estat que per una cosa o per altra l'he anat deixant i tornant a agafar en comptes d'atacar-ho de cop i finalitzar-ho. Durant tot aquest temps, gairebé 6 anys, més dels que vaig trigar en superar tots els crèdits necessaris, he començat a treballar i actualment tinc un lloc de responsabilitat dins de l'empresa, estic vivint amb parella en un pis de propietat, certificacions i formacions diverses, en resum sempre trobant alguna cosa més urgent (que no important) o gratificant a realitzar.

Crec que això ha estat un error per part meva, i suposo que ho he aprés, tal com em va passar durant la fase selectiva, on acostumat a afrontar amb certa tranquil·litat els exàmens de BUP/COU o fins i tot les PAAU, em vaig trobar que no era ben bé el mateix a la universitat.

Tot i això m'ho he passat molt bé fent el projecte ja que la majoria de vegades, no era un problema de temps on escriure molt codi (de fet hi ha poques línies de codi) sinó que cada cosa a realitzar era un repte, que m'ha permès primer entendre com funcionen els sistemes operatius en aspectes no mostrats a cap assignatura del Pla 91 que hagi realitzat, i per d'altra banda a "desenvolupar" un mètode de diagnosi basat en símptomes, molt similar al que actualment està de moda amb la sèrie del "Dr. House". Al no disposar de documentació gaire extensa i molt menys eines que permetessin veure que estava passant dins del sistema operatiu (requeria una versió especial de sistema operatiu (*Checked Build*)) només es podia mitjançant sentinelles i mitjançant els resultats de les proves i el coneixement del funcionament intern dels computadors, sistemes operatius i llenguatges de programació que havia anat adquirint durant la carrera, al final deduir quina podia ser la causa i solucionar-la. Aquest sistema de diagnosi diferencial el continuo aplicant a la vida professional i em permet ser un recurs d'escalat vàlid en un gran numero d'àrees, tot i que en el fons, el meu coneixement dins de l'àrea tecnològica no sigui excessivament profund.

GLOSSARI I REFERÈNCIA

Glossari

- **Shell**: Escriptori. Interfície bàsica d'usuari per a interactuar amb el sistema.
- **Help Desk**: Persona o grup de persones i eines que tracten les incidències de 1º nivell dels usuaris.
- **Unix**: Família de sistemes operatius basats en el System V o BSD. Es caracteritzen per una línia de comandes potent, gran qualitat de la implementació de xarxa i per no ser gaire amigables vers l'usuari.
- **Windows**: Sistema operatiu de Microsoft caracteritzat per una interfície d'usuari fàcil d'utilitzar i que està evolucionant cap a un sistema fiable en entorns interconnectats.
- **Linux**: Sistema operatiu a l'estil de Unix, de codi obert que està tenint una forta presencia mediàtica que n'està accelerant la implantació i evolució.
- **AS/400**: Màquina d'IBM d'arquitectura similar a un Mainframe adaptada per competir amb preu amb servidors Unix o Windows.
- **Mainframe**: Ordinador de gran volum amb microcodi actualitzable, camins d'entrada/sortida millorats, control estricte d'integritat, possibilitat de funcionar en mode degradat (amb components avariats) i oferint múltiples màquines virtuals.
- **SID**: *Security Identifier*. Cadena de bytes que identifiquen unívocament a un usuari de Windows NT de la resta d'usuaris NT del món.
- **Logoff**: Acció de desconnectar-se del sistema.
- **Logon**: Acció d'identificar-se al sistema i accedir a ell.
- **Trojà**: Programa nociu que aparentment realitzarà accions aprovades per l'usuari però a la vegada actuarà de forma oculta per obtenir un profit del privilegis dins del sistema de l'usuari.
- **Debugger**: Aplicació que permet executar pas a pas una altra aplicació donant visibilitat dels continguts de la memòria i del codi a executar. D'aquesta forma permet descobrir els motius de funcionament no conforme amb l'especificació.
- **Espai d'usuari**: Entorn no privilegiat del sistema operatiu on s'executen les tasques de les aplicacions. En cas de necessitar serveis privilegiats, el sistema operatiu comprova per a cada petició si l'aplicació i/o usuari disposen de drets per realitzar l'acció.
- **Service Pack**: Nomenclatura que fa servir Microsoft per indicar subversions de les seves aplicacions. Normalment inclouen millores de funcionalitat i seguretat sobre la versió adquirida.

Referència

[BO01] http://www.codegear.com/products/cppbuilder

[CC02] http://articulos.conclase.net/arboles-b/

[CT01] http://www.citrix.com

[CY01] http://www.cygwin.com/

[DM01] http://www.datamonitor.com

[GG01] http://www.gartner.com

[INE01] http://www.ine.es/inebase/cgi/um?L=&N=&O=pcaxis&M=%2Ft22%2Fp131%2Fa2000t4

[KO01] http://www.kernel.org/pub/linux/libs/pam/modules.html

[LCF01] http://www.fib.upc.es/LCFIB/

[MG01] http://www.metagroup.com

[MIN01] http://www.mingw.org/

[MIT01] http://web.mit.edu/kerberos/www

[MS01] http://msdn.microsoft.com/library/en-us/dllproc/dll_512r.asp

[MS02] http://msdn2.microsoft.com/en-us/library/aa139695.aspx

[MS03] http://msdn2.microsoft.com/en-us/vstudio/default.aspx

[MS04] http://msdn.microsoft.com/library/en-us/security/Security/microsoft_ntlm.asp

[MS05] http://msdn.microsoft.com/library/en-us/security/security/microsoft_kerberos.asp

[MS06] http://www.microsoft.com/sqlserver

[MS07] http://www.microsoft.com/windows2000/server/evaluation/features/dirlist.asp

[MS08] http://www.microsoft.com/technet/scriptcenter/scrguide/sas_reg_fzit.asp

[MS09] http://msdn.microsoft.com/library/en-us/security/security/sspi.asp

[MS10] http://www.microsoft.com/technet/archive/winntas/proddocs/concept/xcp01.mspx

[MS11] http://msdn2.microsoft.com/en-us/library/aa380543.aspx

[MS12] http://msdn2.microsoft.com/en-us/library/aa383015.aspx

[NO01] http://www.novell.com/products/securelogin

[NO02] http://www.novell.com/products/edirectory/

[NS01] http://wp.netscape.com/eng/ssl3/

[OSI01] http://www.opensource.org/licenses/gpl-license.php

[PG01] pGina: Making the big boys play nice

http://pgina.xpasystems.com

[RD01] rdesktop: A Remote Desktop Protocol client

http://www.rdesktop.org

[RFC1050] RPC ftp://ftp.rfc-editor.org/in-notes/rfc1050.txt

[RFC1205] 5250 ftp://ftp.rfc-editor.org/in-notes/rfc1205.txt

[RFC1945] HTTP ftp://ftp.rfc-editor.org/in-notes/rfc1945.txt

[RFC2818] HTTP/TLS ftp://ftp.rfc-editor.org/in-notes/rfc2818.txt

[RFC2865] Radius ftp://ftp.rfc-editor.org/in-notes/rfc2865.txt

[RFC3076] XML http://www.rfc-editor.org/rfc/rfc3076.txt

[RFC4510] LDAP ftp://ftp.rfc-editor.org/in-notes/rfc4510.txt

[SB01] SAMBA - opening windows to a wider world

http://www.samba.org/samba/samba.html

[SM02] Making Login Services Independent of Authentication Technologies

http://java.sun.com/security/jaas/doc/pam.html

[SM01] Solaris Operating System

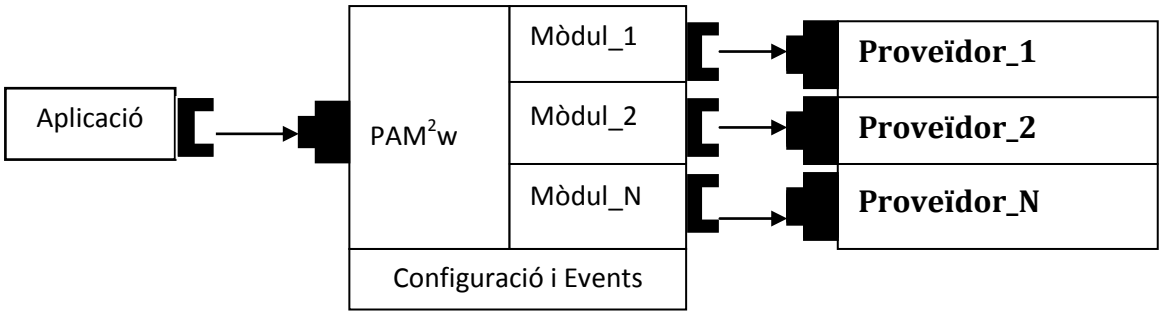
http://www.sun.com/software/solaris/

APÈNDIX A

Ajuda de l'equip de desenvolupament de *software* (SDK) de la PAM^{2}w.

Documentació de les classes disponibles per tal de desenvolupar mòduls

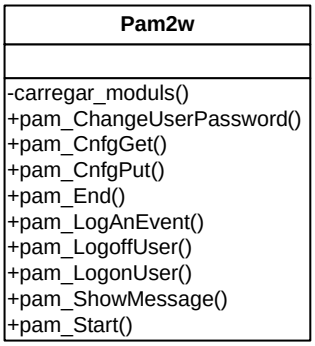
L'esquema de nivells de la plataforma i la comunicació amb la resta dels aplicatius es podria representar de forma semblant a aquesta figura. Tenim una Aplicació que parla amb la PAM^{2}w i aquesta mitjançant la Configuració decideix quin o quins mòduls ha de fer servir. Aquests a la seva vegada establiran la comunicació amb el proveïdor d'autenticació pertinent.



Referència de la Classe PAM2w

Classe on s'implementa la Plataforma PAM2w.

```
#include <PAM2w.h>
```



Documentació de les Funcions membre

void PAM2w::carregar_moduls	**(PAM2wEnv ***	***Env,***
	String	***pszProgName***
	) [private]	

Carrega els mòduls configurats per al programa

void PAM2w::pam_ChangeUserPassword	**(PAM2wEnv ***	***Env,***
	Usuari *	***usuUsuari,***
	String	***pszOldPassword,***
	String	***pszNewPassword***
	)	

L'usuari canvia la contrasenya.

void PAM2w::pam_CnfgGet	**(PAM2wEnv ***	***Env,***
	String	***pszModuleName,***
	String	***pszName,***
	String *	***obValue***
	)	

Recupera un valor de la configuració.

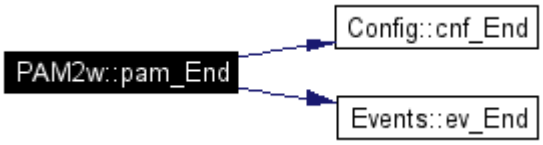
void PAM2w::pam_CnfgPut	**(PAM2wEnv ***	***Env,***
	String	***pszModuleName,***
	String	***pszName,***
	String	***obValue***
	)	

Estableix un valor de la configuració.

void PAM2w::pam_End(PAM2wEnv * ***Env*** **)**

S'ha de cridar al tancar la sessió PAM no quan l'usuari fa logoff.

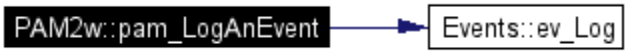
Descarrega de memòria els mòduls i variables



void PAM2w::pam_LogAnEvent	**(PAM2wEnv ***	***Env,***
	String	***pszModuleName,***
	String	***pszMessage,***
	WORD	***dwType,***
	Integer	***dwEventID***
	)	

Afegeix un Succés al visor d'esdeveniments de la màquina local.

Afegeix una entrada en el Log d'esdeveniments del sistema



void PAM2w::pam_LogoffUser	**(PAM2wEnv ***	***Env,***
	Usuari *	***usuUsuari***
	)	

L'usuari del Context vol fer logoff.

Fa logoff de l'usuari al proveïdor que l'ha autenticat

void PAM2w::pam_LogonUser	**(PAM2wEnv ***	***Env,***
	String	***pszUserName,***
	String	***pszPassword,***
	Usuari *	***usuUsuari***
	)	

L'aplicació vol validar un usuari.

Intenta validar l'usuari a un dels múltiples proveïdors configurats

void PAM2w::pam_ShowMessage	**(PAM2wEnv ***	***Env,***
	String	***pszTitle,***
	String	***pszMessage***
	)	

Mostra un missatge a l'usuari.

Demana que l'aplicació mostri un missatge

void PAM2w::pam_Start	**(PAM2wEnv ****	***Env*****,**
	String	***pszProgName*****,**
	ApShowMessage	***CallShowMessage***
	)	

Per iniciar una sessió de Pam.

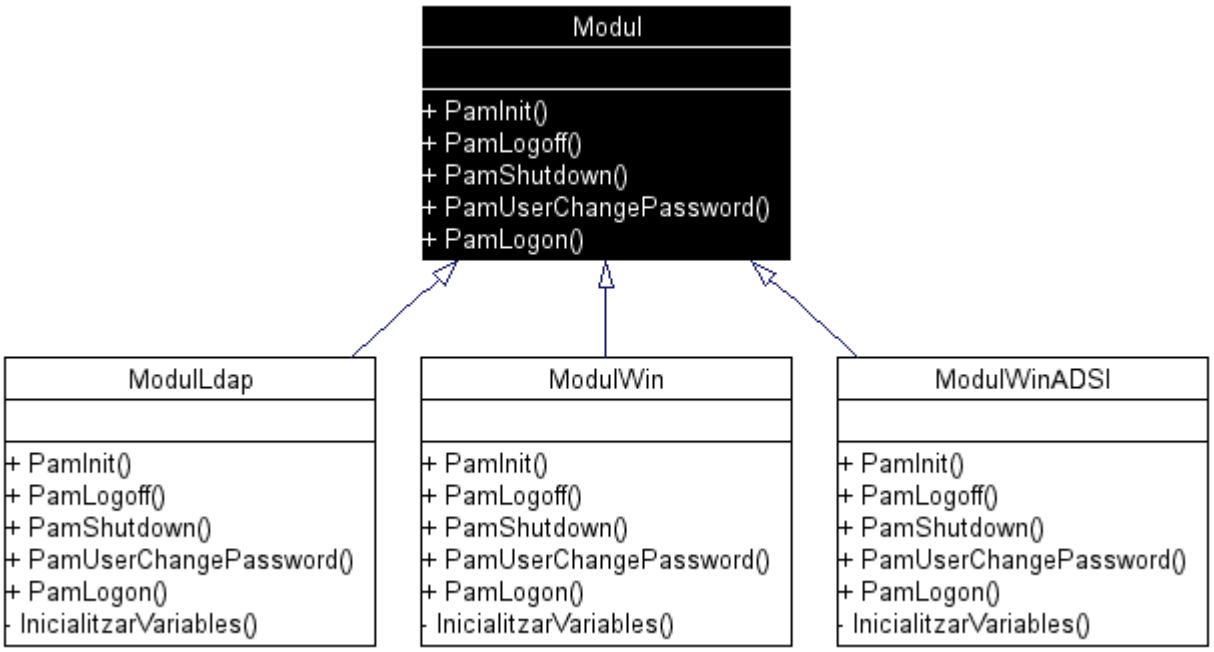
Inicialitza el sistema

Referència de la Classe Modul

Per tal de facilitar el desenvolupament de nous mòduls, s'ha creat una classe que implementa els diversos mètodes necessaris. Llavors al desenvolupar un nou mòdul només hem d'heretar d'aquesta classe i redefinir els mètodes adients, que segurament seran els d'inicialització de l'entorn i de validació de credencials.

```
#include <Modul.h>
```

Diagrama d'Herència per a Mòdul:



Documentació de les Funcions membre

virtual void Modul::PamInit **(Objecte *** ***Env,***
Objecte ***thisPamContext,***
String ***opcions,***
Boolean ***pamPrincipal***
) [virtual]

Es cridada al carregar el Mòdul.

virtual void Modul::PamLogoff(Objecte *context*) [virtual]

És cridada quan l'usuari fa logoff.

```
virtual void Modul::PamLogon          ( Objecte      context,
                                        String       pszUserName,
                                        String       pszPassword,
                                        Usuari *     usuUsuari
                                      ) [virtual]
```

Un usuari vol fer logon, per tant l'hem de validar.

```
virtual void Modul::PamShutdown( Objecte * context ) [virtual]
```

És cridada per tal de descarregar el mòdul de memòria.

```
virtual void Modul::PamUserChangePassword  ( Objecte      context,
                                             String       pszNewPassword,
                                             String       pszOldPassword
                                           ) [virtual]
```

Es cridarà quan l'usuari vulgui canviar la seva contrasenya.

APÈNDIX B

Manual de la PAM2w orientat als administradors de sistemes.

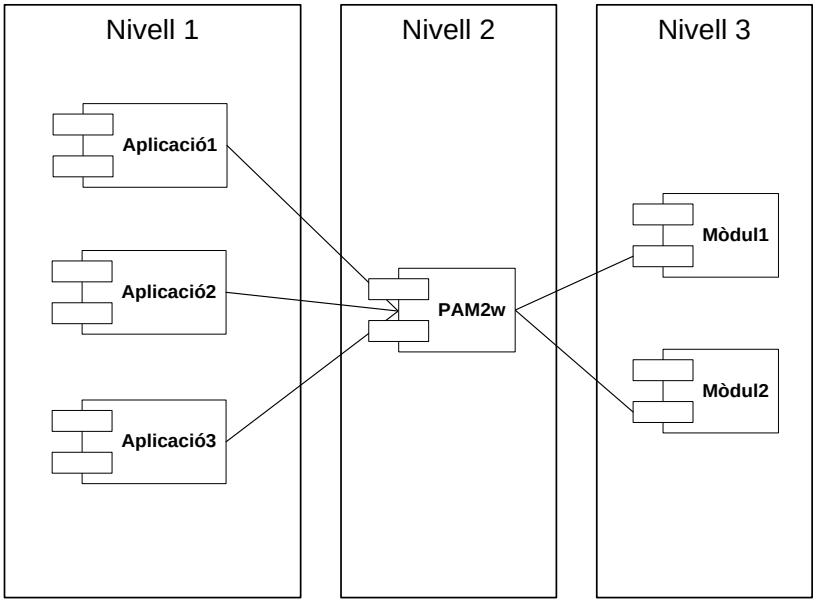
Descripció de la PAM²w

La Plataforma d'autenticació modular per a Microsoft Windows és una peça de programari que permet a les aplicacions compatibles utilitzar proveïdors d'autenticació a discreció de l'administrador de l'entorn, no del desenvolupador de l'aplicació, flexibilitzant d'aquesta forma els mètodes d'accés a les aplicacions per tal que s'adaptin al màxim a les polítiques de seguretat durant l'autenticació implantades. Aquest manual pretén donar una visió bàsica dels components de la plataforma, la seva configuració i com realitzar el diagnòstic en cas de problemes.

Funcionament de la PAM²w

Un primer pas per gestionar aplicacions que utilitzin la Plataforma és comprendre com està dissenyada y els motius d'aquest disseny.

La Plataforma està estructurada en tres nivells, l'aplicació o aplicacions que aprofiten els seus serveis, el nucli i els mòduls que implementen el protocol d'autenticació específic.



Així que en el moment d'engegar l'aplicació, aquesta és la responsable de localitzar els fitxers que contenen el codi del segon nivell i els ha de carregar a memòria. Un cop fet això i havent inicialitzat les dades, el segon nivell buscarà a la configuració quins mòduls ha de carregar per l'aplicació en concret, un cop trobats, els ha d'inicialitzar i aquests realitzar totes les tasques necessàries per establir la comunicació amb el proveïdor d'autenticació implementat.

Configuració de la Plataforma

Per la distribució en tres nivells on els desenvolupadors seran potencialment diferents, la forma d'establir la configuració pot dependre de cada cas, per tant millor llegir el manual de cada un dels components per informació especifica.

En aquest manual incloem informació sobre com configurar la Plataforma i les aplicacions i mòduls d'exemple.

Aplicacions:

Les aplicacions poden accedir al fitxer on s'implementa el codi del nivell 2 mitjançant el camí per defecte definit al sistema operatiu on s'executa (la variable d'entorn PATH) o bé especificant la ruta complerta fins al fitxer. Es recomana accedir al codi de la plataforma estàndard de l'entorn per tal d'aprofitar la reutilització de codi, al oferir la plataforma unes interfícies estables. Tot seguit, comentem les aplicacions que s'ofereixen en la distribució per defecte de la PAM^{2}w.

- **PcqcPAM^{2}w:** Aquest programa no admet configuració ja que només serveix de test de la implantació de la Plataforma. Accepta per línia de comandes un usuari i contrasenya que validarà contra la PAM^{2}w identificant-se com a **Test**
- **PAM^{2}wGiNa**: Aquesta aplicació permet un accés interactiu a Windows utilitzant la Plataforma i per tant permetent l'accés a l'escriptori mitjançant proveïdors d'autenticació arbitraris. En aquest cas es presenta com a **GINA** i es configura directament al registre amb aquests paràmetres dins de "HKLM\Software\Sergi\Gina\Parametres":

Nom	Tipus	Descripció
Admin	DWORD	Si està a 1 indica que si entrem amb un usuari de nom Administrator amb contrasenya correcte a la base de dades d'usuaris locals de Windows, s'ha de permetre l'accés independentment del resultat de la comprovació al proveïdor d'autenticació.
BorrarUsuaris	DWORD	Si està a 1 indica que un cop l'usuari tanqui la sessió hem d'eliminar la compta d'usuari.
CrearUsuaris	DWORD	Si està a 1 indica que si l'usuari disposa d'un accés vàlid segons el proveïdor d'autenticació configurat però no d'un usuari local, es doni d'alta l'usuari local.
Grup	String	Indica a quin grup d'usuaris locals afegirem l'usuari acabat de crear, en cas que l'usuari ja estigui donat d'alta no es mou de grup.
PAM2w	String	Ruta completa fins al fitxer que implementa el nivell dos de la plataforma, el *pam2w.dll.*
TerminalServices	DWORD	Si està a 1 indica que hem de tenir en compte l'entorn com a servidor de terminals i inicialitzar les dades pertinents.
URLDialeg	String	La PAM^{2}wGina mostra un diàleg a l'usuari basat en html, per tant podem fer servir una URL del servidor Web que interessi per mostrar informació actualitzada a l'usuari
ConfiguraDialeg	String	La configuració que ha de fer servir pel diàleg html, com mostrar barres de desplaçament, títol, etc
GinaDLL	String	A "HKLM\software\Microsoft\Windows NT\Current Version\Winlogon" s'ha d'indicar que volem fer servir la Pam2wgina.dll en comptes de la msgina.dll.

Plataforma

El nivell dos de la Plataforma està contingut en un sol fitxer compilat de nom **Pam2w.dll**, com a tal no disposa de configuració però al ser el nexe d'unió entre l'aplicació i el mòdul, per tant disposa de la informació necessària per complir la tasca.

El dipòsit d'informació està a "HKLM\Software\Sergi\PAM2w" i dins d'aquí una carpeta per nom identificatiu d'aplicació. D'aquesta forma es poden definir diversos paràmetres i mòduls a fer servir segons l'aplicació. En tot cas, és recomanable que les aplicacions permetin que l'administrador defineixi la identificació per tal d'estandarditzar al màxim l'entorn.

Nom	Tipus	Descripció
Modul	String	Ruta completa fins al fitxer que implementa el mòdul adequat a l'identificador de l'aplicació.

Mòduls

Els mòduls necessitaran disposar de la configuració que els permeti connectar-se al proveïdor d'autenticació adequat i realitzar les cerques per validar els usuaris. Cada mòdul necessitarà de dades diferents. Anem a descriure els inclosos a la distribució per defecte.

ModulLdap

Aquest mòdul permet la utilització de proveïdors d'autenticació accessibles mitjançant el protocol LDAP.

Nom	Tipus	Descripció
LdapPath	String	URL del servidor LDAP (ldap://nom:port)
LdapdnUser	String	Cadena de cerca a l'LDPA (uid=%s,OU=People,DC=casa)
LdapUseSSL	String	Si 1 indica que ha de fer servir SSL per accedir al servidor LDAP
LdapGroup	String	A quin grup d'usuaris ha d'inicialitzar les variables

ModulWinAdsi

Aquest, fa servir les interfícies ADSI (Active Directory Service Interfaces) per treballar amb el proveïdor d'autenticació de Windows.

Nom	Tipus	Descripció
ADSiNTPath	String	URL del servidor Windows (WinNT://nom_servidor)
ADSiNTdnUser	String	Cadena de cerca
ADSiNTGroup	String	A quin grup d'usuaris ha d'inicialitzar les variables

ModulWin

En canvi aquest fa servir les interfícies històriques de Windows i per tant és compatible amb entorns pre-Windows 2000. En aquest cas, i per exigències del tipus de crida, el procés que faci servir aquest mòdul necessita del privilegi d'actuar com a part del sistema operatiu

Nom	Tipus	Descripció
NTPath	String	Domini Windows on ens validarem
NTdnUser	String	Cadena de cerca
NTGroup	String	A quin grup d'usuaris ha d'inicialitzar les variables